BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

06-020026

CLAIMS

[Claim(s)]
[Claim 1] Character-pattern automatic correction equipment which corrects a location of the alphanumeric alphabetic character concerned to a zero of an alphanumeric character pattern which is characterized by providing the following, and at which an alphanumeric alphabetic character was expressed Said alphanumeric character pattern A detection means to detect a location of the alphanumeric alphabetic character concerned from metrics information which shows physical relationship of a circumscribed quadrangle of the alphanumeric alphabetic character concerned to a zero of this alphanumeric character pattern An operation means to calculate a location of said alphanumeric alphabetic character which should be rearranged from said metrics information A relocation means to rearrange said alphanumeric alphabetic character based on a detection result by said detection means, and the result of an operation by said operation means
[Claim 2] Character-pattern automatic correction equipment which corrects a location of the alphabetic character concerned to a zero of a character pattern which is characterized by providing the following, and at which an alphabetic character was expressed A standard-character pattern with which the standard character was expressed A storage means to memorize metrics information which shows physical relationship of a circumscribed quadrangle of the standard character concerned to a zero of this standard-character pattern Metrics information which shows physical relationship of a circumscribed quadrangle of the alphabetic character concerned to a zero of said character pattern and this character pattern A detection means to detect a location of the standard character corresponding to a location of the alphabetic character concerned, and its alphabetic character based on the contents of storage of said storage means, an operation means to calculate movement magnitude of said alphabetic character based on a detection result by said detection means, and a relocation means that rearranges said alphabetic character based on the result of an operation by said operation means

[Translation done.]

* NOTICES *

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]

[Industrial Application] This invention relates to the character-pattern automatic correction equipment which makes the automatic correction of the location of an alphabetic character.
[0002]

[Description of the Prior Art] In the former, it is possible by creating a character pattern newly by the user side, and registering with a system to use the alphabetic character. Locations and alphabetic character lists, such as the base line, are also taken into consideration and created by the user, and the character pattern also has many artificial mistakes. Moreover, the character pattern may be created, without taking an alphabetic character list into consideration at all.

[0003] As what inspects a character pattern, what was indicated by JP,60-29783,A and JP,3-176764,A is known that such a problem should be solved.

[0004] What was indicated by JP,60-29783,A displays a creation character pattern and a standard-character pattern side by side on the same screen, or puts in order and prints them on the same space, and inspects both character patterns visually.

[0005] Moreover, the base line about the capital letter of the alphabet judges whether it is the right by what was indicated by JP,3-176764,A detecting the location of the base line from an alphabetic character image by detecting the Rhine location of the bottom of the dot pattern which constitutes the capital letter of the alphabet, and comparing this location and reference value that were detected.

[0006]
[Problem(s) to be Solved by the Invention] However, since it limited only to a design font inspection, it was unsuitable for location inspection of the base line, Capheight, Xheight, etc., etc. what was indicated by above-mentioned JP,60-29783,A.

[0007] Moreover, in what was indicated by above-mentioned JP,3-176764,A, since the amount of adjustments by the design-technique for the vision amendment called an overhang (overhang) is not taken into consideration, an exact character pattern cannot be

obtained. Moreover, the inspection about the capital letter of the alphabet cannot be checked about alphabetic characters, such as a small letter of the alphabet, a numeric character, ** kanas, and katakana, although it is possible. That is, since it limits to the check of the dot pattern of the capital letter of the alphabet, it is impossible to check the location of a certain character pattern of about 7000 characters like JIS.

[0008] This invention aims at offering the character-pattern automatic correction equipment which can correct the location of character patterns, such as an alphanumeric alphabetic character and ** kanas, to the optimal location.

[0009]
[Means for Solving the Problem] In order to attain the above-mentioned purpose, the 1st invention is character-pattern automatic correction equipment which corrects a location of the alphanumeric alphabetic character concerned to a zero of an alphanumeric character pattern at which an alphanumeric alphabetic character was expressed. Said alphanumeric character pattern, A detection means to detect a location of the alphanumeric alphabetic character concerned from metrics information which shows physical relationship of a circumscribed quadrangle of the alphanumeric alphabetic character concerned to a zero of this alphanumeric character pattern, It has an operation means to calculate a location of said alphanumeric alphabetic character which should be rearranged from said metrics information, and a relocation means to rearrange said alphanumeric alphabetic character based on a detection result by said detection means, and the result of an operation by said operation means.

[0010] Moreover, a standard-character pattern with which the 2nd invention is character-pattern automatic correction equipment which corrects a location of the alphabetic character concerned to a zero of a character pattern at which an alphabetic character was expressed, and the standard character was expressed, A storage means to memorize metrics information which shows physical relationship of a circumscribed quadrangle of the standard character concerned to a zero of this standard-character pattern, Metrics information which shows physical relationship of a circumscribed quadrangle of the alphabetic character concerned to a zero of said character pattern and this character pattern, A detection means to detect a location of the standard character corresponding to a location of the alphabetic character concerned, and its alphabetic character based on the contents of storage of said storage means, It has an operation means to calculate movement magnitude of said alphabetic character based on a detection result by said detection means, and a relocation means which rearranges said alphabetic character based on the result of an operation by said operation means.

[0011]
[Function] With the character-pattern automatic correction equipment of the 1st invention The location of the alphanumeric alphabetic character concerned is detected from the metrics information which shows the physical relationship of the circumscribed quadrangle of the alphanumeric alphabetic character concerned to the zero of an alphanumeric character pattern and this alphanumeric character pattern with a detection means. With moreover, an operation means The location of the alphanumeric alphabetic

character which should be rearranged from said metrics information calculates, and said alphanumeric alphabetic character is further rearranged by the relocation means based on the detection result by said detection means, and the result of an operation by said operation means. Therefore, the automatic correction of the location of the alphanumeric character pattern of the capital letter and small letter of the alphabet, or a numeric character can be made in the optimal location.

[0012] moreover, with the character-pattern automatic correction equipment of the 2nd invention The metrics information which shows the physical relationship of the circumscribed quadrangle of the standard character concerned to the zero of the standard-character pattern with which the standard character was expressed, and this standard-character pattern to a storage means is memorized. With a detection means The metrics information which shows the physical relationship of the circumscribed quadrangle of the alphabetic character concerned to the zero of said character pattern and this character pattern, Based on the contents of storage of said storage means, the location of the standard character corresponding to the location of the alphabetic character concerned and its alphabetic character is detected. With moreover, an operation means The movement magnitude of said alphabetic character calculates based on the detection result by said detection means, and said alphabetic character is further rearranged by the relocation means based on the result of an operation by said operation means. Therefore, the automatic correction of the location of the character pattern corresponding to the character code of JIS, such as the capital letter and small letter of the alphabet, and a numeric character, a ** kana alphabetic character, can be made in the optimal location.

[0013]
[Example] Hereafter, the 1st example and 2nd example of this invention are explained with reference to an accompanying drawing.

[0014] The 1st example is first explained with reference to drawing 1 thru/or drawing 9 .

[0015] Drawing 1 shows one example of the character-pattern automatic correction equipment concerning this invention by the functional block diagram.

[0016] In this drawing character-pattern automatic correction equipment The input unit 10 which inputs the character code corresponding to a character font, and the font memory 20 which stores the alphanumeric character pattern, The reading section 30 which reads the character pattern corresponding to the specified character code from a font memory 20, The detecting element 40 which detects the location of each alphabetic character, and the Rhine operation part 50 which asks for the list line (Rhine) of an alphanumeric alphabetic character, It has the arrangement operation part 60 which calculates these errors (movement magnitude) in quest of the detected location and the location which should be moved, and the relocation section 70 which rearranges an alphabetic character based on the movement magnitude calculated by the arrangement operation part 60, and is constituted.

[0017] Here, the pattern information which consists of the alphanumeric character pattern (the outline or bit map) and metrics information corresponding to a character code and its character code on ASCII (ASCII) is stored in the font memory 20.

[0018] This pattern information is explained taking the case of the one outline "7" shown in drawing 2 .

[0019] Metrics information has six control points C which are intersections of each straight line for expressing the outline "7", the bounding box (maximum rectangle frame which touches character pattern) information surrounded by the drawing 2 middle point line, and the information on the escapement W which is the delivery width of face of an alphabetic character, and is constituted. Bounding box information consists of four information, the ascender A (positive value) which is an offset value from a zero (0 0), Descender D (positive value), the light bearing R (positive value), and the left bearing L (negative value).

[0020] Moreover, there are some list lines (Rhine) in principle in an alphanumeric alphabetic character, and, as for each alphabetic character, it is common to have stood in a line along with those lines. Each of those Rhine is explained with reference to drawing 3 .

[0021] In this example, eight Rhine LN1, LN2, LN3, LN4, LN5, LN6, LN7, and LN8 is set up.

[0022] LN2 shows a cap line (cap line), LN4 shows a MIN line (mean line), LN1 shows an ascender line (ascender line), and LN7 shows [LN5 shows the base line (base line), and] the descender line (decender line). The ascender line LN1 is the highest Rhine of the small letter of the alphabet, and the descender line LN7 is the lowest Rhine of the small letter of the alphabet here.

[0023] Rhine LN3, LN6, and LN8 is Rhine required in order to ask for an overhang (overhang). An overhang is adjustment by the design-technique for vision amendment here. There are four kinds of these overhangs, a cap overhang (cap overhang), a MIN overhang (mean overhang), a base overhang (base overhang), and a descender overhang (decender overhang), and it can ask for these as follows.

[0024]
Cap overhang = value of value-Rhine LN2 in Rhine LN1 -- (1)
MIN overhang = value of LN3 - Value of LN4 -- (2)
Base overhang = value of LN5 - Value of LN6 -- (3)
Descender overhang = value of LN7 - Value of LN8 -- (4)
These overhangs are calculated by the Rhine operation part 50.

[0025] In this example, since the value of the base line LN5 is set to Y coordinate value =0, each value of ascender Rhine LN1, cap Rhine LN2, Rhine LN3, MIN Rhine LN4, and the base line LN5 turns into a positive value, on the other hand each value of Rhine

LN6, descender Rhine LN7, and Rhine LN8 turns into a negative value.

[0026] In addition, the detection means mentioned above has a detecting element 40, and it is constituted, the operation means mentioned above has the Rhine operation part 50 and the arrangement operation part 60, and it is constituted, and the relocation means mentioned above has the relocation section 70, and consists of this example.

[0027] Next, an example is given and location correction of a character pattern is explained.

[0028] If the sequential input of the character code corresponding to each alphabetic character of an alphabetic character "A", "n", "i", and "O" is carried out from an input unit 10, respectively, the read-out section 20 will read a font file, and will read the character pattern corresponding to those character codes out of the file from a font memory 20. Consequently, the list of an alphanumeric character pattern as shown in drawing 4 will be obtained. However, it turns out that these alphanumeric character patterns are not the optimal lists as shown in drawing 4 . In addition, in this example, the design of an alphanumeric character pattern is taken as a right thing as a prerequisite.

[0029] Then, since it is necessary to inspect the location of those character patterns and to correct to the optimal location next, the inspection and the correction of the location of a character pattern concerning this invention are explained.

[0030] Here, it will explain taking the case of the first alphabetic character "A", and the processing process which inspects and corrects the location of the alphabetic character is explained using drawing 5 .

[0031] A font file is read in the reading section 30, and an alphabetic character is distinguished by the character code. In this case, since it is an ASCII code, "A" of a capital letter is Ox41. That is, the reading section 30 will judge with it being "A" of a capital letter, if the character code Ox41 from an input unit 10 is received.

[0032] Here, the patterns of "A" of a capital letter shall be contents as shown in drawing 5 (a), and the bounding box shall exist in the location where it existed on the left of the center of Escapement W (it goes to space and is left-hand side), and the base of a bounding box touched descender-Rhine.

[0033] In a detecting element 40, the center-of-gravity location is detected from the bounding box information on a capital letter "A." Here, let the intersection of the diagonal line of a bounding box be a center-of-gravity location. Specifically, a bounding box can express the point shown, respectively with the value of the point shown, respectively with the value of the light bearing in descender Rhine, and the value of left bearing, and the light bearing in ascender Rhine, and the value of left bearing in the rectangle field made into top-most vertices. Therefore, it can ask for the intersection of the diagonal line based on these values.

[0034] It asks for each Rhine as shown in drawing 3 in the Rhine operation part 50 (about the details of this operation, it mentions later).

[0035] In the arrangement operation part 60, it investigates whether the coordinate value (X0, Y0) which shows a center-of-gravity location exists in the center of Escapement W. If it does not exist in a center, the movement magnitude of the direction of X is calculated. In this case, it becomes movement magnitude XM=W-X0.

[0036] Thus, if the movement magnitude of the direction of X is calculated, in the arrangement operation part 60, the movement magnitude of the direction of Y will be calculated next. Since it is the location where the condition that the base line LN5 shown in drawing 3 touched the base of a bounding box is suitable in "A", the value of the descender to this alphabetic character is "0." However, in this example, since the base of a bounding box is in contact with the descender line, the movement magnitude of the direction of Y in this case is a descender D value.

[0037] In the relocation section 70, only d moves in the direction of X in XM and the direction of Y. Thereby, as a bounding box is shown in drawing 5 (b), only XM is moved in the direction of X, and further, as shown in drawing 5 (b), only a descender D value is moved in the direction of Y. In addition, the sequence is not asked in the case of migration in the direction of X and the direction of Y based on movement magnitude.

[0038] About other alphabetic characters in which the base of a bounding box touches the base line like a capital letter "A" as mentioned above, the location of an alphabetic character is correctable by the same processing.

[0039] Also about alphabetic characters other than such an alphabetic character, it can ask by the method same about the movement magnitude of the direction of X as the above, and on the other hand, about the movement magnitude of the direction of Y, if it turns out whether the base of a bounding box touches either of Rhine LN5, LN6, LN7, and LN8, it can ask.

[0040] Generally, both the alphabet {A, B, D, E, F, H, I, K, L, M, N, P, R, T, V, W, X, Y, Z} and a numeric character {1, 2, 4, 7} touch the base line LN5 which the base of a bounding box showed to drawing 3 .

[0041] Moreover, both the alphabet {C, G, O, Q, S, U} and a numeric character {0, 3, 6, 8, 9} touch Rhine LN6 which the base of a bounding box showed to drawing 3 .

[0042] On the other hand, since the small letter of the alphabet can be classified into four kinds of a short letter and ascender letter, a descender letter, and a long letter according to height and the list of Rhine, it serves as arrangement according to the class.

[0043] Among short letters, both {a, c, e, o, s, and u} touch Rhine LN6 which the base of a bounding box showed to drawing 3 , and both {m, n, r, v, wx, and z} touch the base line LN5 which the base of a bounding box showed to drawing 3 .

[0044] Among ascender letters, both {b, d, and t} touch Rhine LN6 which the base of a bounding box showed to drawing 3 , and both {h, i, k, and l} touch the base line LN5 which the base of a bounding box showed to drawing 3 .

[0045] Among descender letters, both {p and q} touch the descender line LN7 which the base of a bounding box showed to drawing 3 , and both {g and y} touch Rhine LN8 which the base of a bounding box showed to drawing 3 .

[0046] A long letter {j} touches Rhine LN8 which the base of a bounding box showed to drawing 3 .

[0047] In this example, the small letter of the alphabet shall follow the regulation mentioned above.

[0048] Next, how to ask for each Rhine other than the base line by the Rhine operation part 50 is explained with reference to drawing 6 and drawing 7 .

[0049] Alphabetic character"H", "O", "x", and "p" surely consider as a certain thing as a prerequisite here at the font to input. In addition, the alphabetic character in which the alphabetic character to input is not limited to these alphabetic characters, and the base of a bounding box touches the base line LN5 like alphabetic character"H", An alphabetic character "O" and the alphabetic character in which the base of a bounding box touches Rhine LN6 similarly, You may make it input the alphabetic character corresponding to the alphabetic character in which the base of a bounding box touches the base line LN5 like short RETA "x", and the alphabetic character in which the base of a bounding box touches the descender line LN7 like a descender letter "p", respectively.

[0050] In addition, in drawing 6 and drawing 7 , "cap overhang" shows a cap overhang, "mean overhang" shows a MIN overhang, "BBox" shows a bounding box and "descender overhang" shows ["overhang" shows an overhang and / "base overhang" shows a base overhang and] the descender overhang.

[0051] First, overhang (it is assumed that it is cap overhang=mean overhang =base overhang =descender overhang) is calculated. This data processing is explained with reference to the flow chart shown in drawing 6 .

[0052] The Rhine operation part 50 reads a character code (step 601), and judges whether it is the character code of "H" of a capital letter (step 602). When it is the character code of "H" of a capital letter, the metrics information on "H" of a capital letter is read (step 603), and the height of the bounding box BBox of the alphabetic character is found (step 604).

[0053] The height of the bounding box BBox is obtained by adding the value of an ascender, and the value of descender. Since the value of this ascender and the value of descender are included in the metrics information on "H" of the capital letter read now, they can be easily known from that metrics information.

[0054] Moreover, when it is not the character code of "H" of a capital letter in the above-mentioned step 602, it judges next whether it is the character code of "O" of a capital letter (step 605).

[0055] When it is the character code of "O" of a capital letter, the metrics information on "O" of a capital letter is read (step 606), and the height of the bounding box BBox of the alphabetic character is found (step 607). The height of the bounding box BBox in this case as well as the above can be found.

[0056] Thus, if the height of "H" of a capital letter and the bounding box BBox of "O" is found, the Rhine operation part 50 will ask for an overhang by calculating a degree type (step 608).

[0057]
overhang= (height of BBox of height-"H" of BBox of "O")/2 -- (5)
The value of the overhang called for by calculating this formula (5) serves as a cap overhang and a base overhang.

[0058] When it is not the character code of "O" of a capital letter in the above-mentioned step 605, it judges whether it is the character code of "x" of a small letter (step 609).

[0059] When it is the character code of "x" of a small letter, the metrics information of "x" on a small letter is read (step 610), and the height of the bounding box BBox of the alphabetic character is found like the above (step 611).

[0060] On the other hand, when it is not the character code of "x" of a small letter in step 609, it judges whether it is the character code of "p" of a small letter (step 612).

[0061] In the case of the character code of "p" of a small letter, the metrics information on "p" of a small letter is progressed to step 611 reading ****** (step 613) and after that.

[0062] In addition, when it is not the character code of "p" of a small letter in step 612, it returns to step 601 and this step or subsequent ones is performed.

[0063] Next, decision processing of each above-mentioned Rhine by the Rhine operation part 50 is explained with reference to the flow chart shown in drawing 7 .

[0064] First, the Rhine operation part 50 defines it as cap overhang=mean overhang =base overhang =descender overhang (step 701), and it asks for Rhine LN1, LN2, LN3, LN4, LN6, LN7, and LN8 based on these contents of a definition, and the value of the base line set up beforehand. He is trying to specifically ask for each Rhine as follows.

[0065] First, it can ask for Rhine LN6 by subtracting the value of a base overhang from the value of the base line LN5 (step 702 (the value of the value-base overhang of the Rhine LN6= base line LN5)). The value of this Rhine turns into a negative value.

[0066] Next, it can ask for cap Rhine LN2 by adding the value of the base line LN5, and the height of BBox of "H" (step 703 (height of BBox of value +"H" of the cap Rhine LN2= base line LN5)). The value of this Rhine turns into a positive value.

[0067] In this way, it can ask for ascender Rhine LN1 by adding the value of cap Rhine LN2 and the value of a cap overhang which were acquired (step 704 (the value of the value + cap overhang of ascender Rhine LN1= cap Rhine LN2)). The value of this Rhine turns into a positive value.

[0068] It can ask for MIN Rhine LN4 by adding with the value of the base line LN5, and the height of BBox of a small letter "x" (step 705 (height of BBox of value [of the MIN Rhine LN4= base line LN5] + "x")). The value of this Rhine turns into a positive value.

[0069] In this way, it can ask for Rhine LN3 by adding the value of MIN Rhine LN4 and the value of a MIN overhang which were calculated (step 706 (the value of the value + MIN overhang of Rhine LN3= MIN Rhine LN4)). The value of this Rhine turns into a positive value.

[0070] Moreover, descender Rhine LN7 can be calculated by subtracting the height of BBox of a small letter "p" from the value of LN3 calculated in this way (step 707 (value of descender Rhine LN7= Rhine LN3 - height of BBox of "p")). The value of this Rhine turns into a negative value.

[0071] Furthermore, Rhine LN8 can be calculated by subtracting the value of a descender overhang from the value of descender Rhine LN7 called for in this way (step 708 (the value of the value-descender overhang of Rhine LN8= descender Rhine LN7)). The value of this Rhine turns into a negative value.

[0072] Since each Rhine was determined by the Rhine operation part 50 by the above processing, based on the value of each of those Rhine, the movement magnitude of the direction of Y of each alphabetic character can be calculated.

[0073] Next, data processing of the movement magnitude of the direction of Y of each alphabetic character by the arrangement operation part 60 is explained with reference to the flow chart shown in drawing 8 .

[0074] The arrangement operation part 60 reads a character code (step 801), and judges whether it is a character code corresponding to the alphabetic character of either of the alphabetic character groups shown below (step 802).

[0075] That is, it judges whether it is the character code of one alphabetic character of the capital letter {A, B, D, E, F, H, I, K, L, M, N, P, R, T, V, W, X, Y, Z} of the alphabet, a numeric character {1, 2, 4, 7}, and the small letter {m, n, r, v, wx, z, h, i, k, l} of the alphabet.

[0076] Here, when it is one of character codes, the movement magnitude to the direction of Y is calculated so that the base of the bounding box BBox of the alphabetic character may touch the base line LN5 (step 803).

[0077] When it is not the character code of the alphabetic character which corresponds in step 802 when step 803 is ended, it judges whether it is a character code corresponding to the alphabetic character of either of the alphabetic character groups shown below (step 804).

[0078] That is, it judges whether it is the character code of the alphabetic character of either the capital letter {C, G, O, Q, S, U} of the alphabet, a numeric character {0, 3, 6, 8, 9} and the small letter {a, c, e, o, s, u, b, d, t} of the alphabet.

[0079] Here, when it is one of character codes, the movement magnitude to the direction of Y is calculated so that the base of the bounding box BBox of the alphabetic character may touch Rhine LN6 (step 805).

[0080] When it is not the character code of the alphabetic character which corresponds in step 804 when step 805 is ended, it judges whether it is a character code corresponding to the alphabetic character of either of the alphabetic character groups shown below (step 806).

[0081] That is, it judges whether it is the character code of one alphabetic character of the small letters {p, q} of the alphabet.

[0082] Here, when it is one of character codes, the movement magnitude to the direction of Y is calculated so that the base of the bounding box BBox of the alphabetic character may touch descender Rhine LN7 (step 807).

[0083] When it is not the character code of the alphabetic character which corresponds in step 806 when step 807 is ended, it judges whether it is a character code corresponding to the alphabetic character of either of the alphabetic character groups shown below (step 808).

[0084] That is, it judges whether it is the character code of one alphabetic character of the small letters {g, y, j} of the alphabet.

[0085] Here, when it is one of character codes, the movement magnitude to the direction of Y is calculated so that the base of the bounding box BBox of the alphabetic character may touch Rhine LN8 (step 809).

[0086] It judges whether when it is not the character code of the alphabetic character which corresponds in step 808 when step 809 is ended, the alphabetic character which should be processed exists (step 810), in existing, it returns to the above-mentioned step 801, and performs this step or subsequent ones, and on the other hand, when it does not exist, processing is ended.

[0087] In this way, based on the calculated movement magnitude, an alphabetic character is rearrangeable in the optimal location.

[0088] Next, relocation processing of the alphabetic character by the relocation section 70 is explained with reference to the flow chart shown in drawing 9 .

[0089] The relocation section 70 calculates a coordinate value as movement magnitude of the direction of X of a character, and the direction of Y. Here, it considers as the movement magnitude which considered as the movement magnitude calculated by the detecting element 40 as movement magnitude of the direction of X, and defined this as Xm1, and was calculated by the arrangement operation part 60 as movement magnitude of the direction of Y, and this is defined as Ym2 (step 901).

[0090] If movement magnitude (Xm1, Ym2) is calculated, each value about the bounding box BBox will be changed.

[0091] Namely, about an ascender and descender, only Ym2 moves in the direction of Y, respectively, and only Xm1 moves in the direction of X about left bearing and light bearing, respectively (step 902).

[0092] Next, as for the relocation section 70, the target character pattern judges whether it is expressed by the bit map (step 903).

[0093] If it is a bit map, only Xm1 and Ym2 will move the coordinate point (xi, yi) that a dot exists. that is, the coordinate point of n dots -- then (i(xi, yi) = 1, 2, -- n (integer)), a dot is moved to the coordinate which calculates each type of xi=xi+Xm1 and yi=yi+Ym2, and is acquired (step 904).

[0094] the case (in this case, it will be called outline) where it is not a bit map in step 903 -- the coordinate point of n control points -- then (i(xi, yi) = 1, 2, -- n (integer)), a dot is moved to the coordinate which calculates each type of xi=xi+Xm1 and yi=yi+Ym2, and is acquired (step 905).

[0095] When step 904 is ended and step 905 is ended, it judges whether the alphabetic character which should be processed exists (step 906).

[0096] Here, in existing, it returns to the above-mentioned step 901, and performs this step or subsequent ones, and on the other hand, when it does not exist, processing is ended.

[0097] It means that the automatic correction of the location of an alphanumeric character pattern had been made by the above processing in the optimal location.

[0098] Next, the 2nd example of this invention is explained with reference to drawing 10 thru/or drawing 17 .

[0099] Drawing 10 shows the 2nd example of the character-pattern automatic correction equipment concerning this invention by the functional block diagram.

[0100] In this drawing character-pattern automatic correction equipment The input unit 1010 which inputs the character code corresponding to a character font, The font memory 1020 which stores the created character pattern, The reading section 1030 which reads the character pattern corresponding to the specified character code from a font memory 1020, While detecting the location of the specifications object font memory 1040 which stores the character pattern of the specifications object set up beforehand, and the character pattern read by the reading section 1030 The detecting element 1050 which reads the character code of that character pattern, and the character pattern of a corresponding specifications object from the specifications object font memory 1040, and detects the location of this alphabetic character Batang, It has the arrangement operation part 1060 which calculates the location which the alphabetic character created based on the location of a character pattern and the location of the character pattern of a specifications object which were created should move, and the relocation section 1070 which rearranges an alphabetic character in the location called for by the arrangement operation part 1060, and is constituted.

[0101] The pattern information which consists of the character pattern (the outline or bit map) and metrics information corresponding to a character code and its character code on JIS is stored in the font memory 20. Here, the created character pattern as shown in drawing 11 (a) and (b) is stored. Here, the design of an alphabetic character is taken as a right thing as a prerequisite.

[0102] The character pattern of the specifications object used as the setmaster of all typefaces is stored in the specifications object font memory 1040, and, specifically, the pattern information which consists of the character pattern (the outline or bit map) and metrics information corresponding to a character code and its character code on JIS is stored like the above. Here, the character pattern of a specifications object as shown, for example in drawing 12 , for example is stored. Here, it is JIS, for example. 6877 characters of X0208 shall be stored and the design of a specifications object and the list of an alphabetic character shall be right data as a prerequisite.

[0103] Here, the above-mentioned pattern information is explained taking the case of "A" shown in drawing 13 .

[0104] Metrics information has the alphabetic character mask "A", the information on the bounding box BBox (the maximum rectangle frame which touches a character pattern) surrounded by the drawing 13 middle point line, and the information on the escapement W which is the delivery width of face of an alphabetic character, and is constituted. Bounding box BBox information consists of four information, the ascender A (positive value) which is an offset value from a zero (0 0), Descender D (negative value), the light bearing R (positive value), and the left bearing L (negative value). Each character pattern of the character pattern and specifications object with which such metrics information

was created is defined.

[0105] In addition, the storage means mentioned above has the specifications object font memory 1040, and it consists of this example, the detection means mentioned above has a detecting element 1050, and it is constituted, the operation means mentioned above has the arrangement operation part 1060, and is constituted, and a relocation means has the relocation section 1070 and is constituted.

[0106] In the starting configuration, location correction processing of the character pattern of character-pattern automatic correction equipment is explained.

[0107] the alphabetic character "A" from an input unit 1010, "B", and "**" -- " -- if it is and the sequential input of the character code corresponding to each alphabetic character of "is carried out, respectively, the read-out section 1030 will read a font file from a font memory 1020, and will read the character pattern corresponding to those character codes from the inside of the file. Consequently, the list of a character pattern as shown in drawing 11 (a) will be obtained. However, it turns out that these character patterns are not the optimal lists as shown in drawing 12 .

[0108] Then, since it is necessary to inspect the location of those character patterns and to correct to the optimal location next, the inspection and the correction of the location of a character pattern concerning this invention are explained.

[0109] Here, it will explain taking the case of the first alphabetic character "A", and the processing process which inspects and corrects the location of the alphabetic character is explained with reference to drawing 14 thru/or drawing 16 .

[0110] A font file is read in the reading section 1030, and an alphabetic character is distinguished by the character code. In this case, since it is JIS code, "A" of a capital letter is Ox2341 (JIS3 division 33 point). That is, if the character code Ox2341 from an input unit 1010 is received, the reading section 1030 judges with it being "A" of a capital letter based on this character code, and it will pass these information and character codes that were read further to a detecting element 1050 while it reads the character pattern and metrics information on the alphabetic character in a font file.

[0111] In a detecting element 1050, y0, y1, x0, and x1 which are shown in drawing 14 are detected from the bounding box information (an ascender, descender, left bearing, light bearing) on an alphabetic character "A." In drawing 14 , as for the length of the length of the field (body size) where, as for MX, an alphabetic character is registered and which was set up beforehand, and MY, the length beside said field (body size) is shown.

[0112] That is, y0 is the same as that of the absolute value of the value of descender.
(y0=- (value of descender))
y1 is a value which subtracts the value of an ascender and is acquired from the value of MY.
(Value of the value-ascender of y1=MY)

x0 is the same ****** as the absolute value of the value of left bearing.
(x0=- (value of left bearing))
x1 is a value which subtracts the value of light bearing and is acquired from the value of MX.
(Value of the value-light bearing of x1=MX)
Thus, the value of y0, y1, x0, and x1 which were calculated is defined as positional information about the created character pattern.

[0113] Next, a detecting element 1050 reads the alphabetic character of a specifications object "A", i.e., the alphabetic character of a character code Ox2341, from the specifications object font memory 1040. Here, the alphabetic characters "A" of a specifications object shall be contents as shown in drawing 15 . In drawing 15 , MX0 shows the length of the length of body size, and MY0 shows the length beside body size. Also about this standard-character pattern, ya, yb, xa, and xb which are shown in drawing 15 are detected from the bounding box information on the alphabetic character "A" of a specifications object. These values can be calculated like the above.

[0114] Namely, value yb= of the value-ascender of ya=MY0 - (value of descender)
xa= - (value of left bearing)
the value of the value-light bearing of xb=MX0 -- the value of ya, yb, xa, and xb which were calculated by doing in this way is defined as positional information about the character pattern of a specifications object.

[0115] The positional information about the created character pattern and the character pattern of a specifications object is outputted to the arrangement operation part 1060 from a detecting element 1050.

[0116] A character pattern as shown in drawing 16 as a result will be obtained by the arrangement operation part's 1060 comparing the positional information about the created character pattern, and the positional information about the character pattern of a specifications object, calculating the movement magnitude to the optimal location based on this comparison result, and moving further the character pattern created by the relocation section 1070 based on that movement magnitude.

[0117] Next, data processing of the movement magnitude by the arrangement operation part 1060 is explained with reference to the flow chart shown in drawing 17 .

[0118] The arrangement operation part 1060 defines it as A= (x0+x1), and defines it as B= (y0, y1) while it reads the positional information (y0, y1, x0, x1) about the character pattern created from the detecting element 1050 (step 1701). Moreover, the positional information (ya, yb, xa, xb) about the character pattern of the specifications object from a detecting element 1050 is read (step 1702).

[0119] Since the location of the direction of X will be right when the arrangement operation part 1060 investigates whether the formula of x0:x1=xa:xb is materialized (step 1703) and this formula is materialized, next, the movement magnitude about the direction

of Y will be calculated.
[0120] On the other hand, since the location of the alphabetic character of the direction of X is not the optimal in step 1703 when the above-mentioned formula is abortive, movement magnitude is computed that the alphabetic character should be moved to the optimal location (step 1704).

[0121] namely, -- x0 new:x1 new=xa:xb -- (6)
Each value of x0new and x1new is computed so that a ** type may be materialized.

[0122] In this case, it is from the relation of value-(value of value + left bearing of light bearing) =A (constant) of x0+x1=x0 new+x1 new=MX. x1 new=A-x0new -- (7)
A ** type is obtained.

[0123] From the relation of this formula (7) to moreover, the above-mentioned formula (6) x0new:(A-x0new) =xa:xb -- (8)
It can deform into a ** type.

[0124] Furthermore, since A, xa, and xb are constants in a formula (8), it is from a formula (8). x0new=(Axxa)/(xa+xb) -- (9)
A ** type is obtained.

[0125] The optimal location of the value of X of x0new, i.e., the direction, is obtained by calculating this formula (9).

[0126] And the arrangement operation part 1060 makes the value which subtracted x0 from the value of x0new which calculated the above-mentioned formula (9) and was obtained the movement magnitude of the direction of X (step 1705). (movement magnitude of the direction of X = x0 new-x0)

[0127] Moreover, when the arrangement operation part 1060 investigates whether the formula of y0:y1=yb:ya is materialized when step 1705 is ended and a formula is materialized in the above-mentioned step 1703 (step 1706), and this formula is materialized, the location of the direction of Y will be right.

[0128] On the other hand, since the location of the alphabetic character of the direction of Y is not the optimal in step 1706 when the above-mentioned formula is abortive, movement magnitude is computed that the alphabetic character should be moved to the optimal location (step 1707).

[0129] namely, -- y0 new:y1 new=yb:ya -- (10)
Each value of y0new and y1new is computed so that a ** type may be materialized.

[0130] In this case, it is from the relation of value-(value of value + descender of ascender) =B (constant) of y0+y1=y0 new+y1 new=MY. y1 new=B-y0new -- (11)
A ** type is obtained.

[0131] From the relation of this formula (11) to moreover, the above-mentioned formula (10) y0new:(B-y0new) =yb:ya -- (12)
It can deform into a ** type.

[0132] Since B, ya, and yb are furthermore constants in a formula (12), it is from a formula (12). y0new=(Bxyb)/(ya+yb) -- (12)
A ** type is obtained.

[0133] The optimal location of the value of Y of y0new, i.e., the direction, is obtained by calculating this formula (12).

[0134] And the arrangement operation part 1060 makes the value which subtracted y0 from the value of y0new which calculated the above-mentioned formula (12) and was obtained the movement magnitude of the direction of Y (step 1708). (movement magnitude of the direction of Y = y0 new-x0)

[0135] Processing is ended when step 1708 is ended, and a formula is materialized in the above-mentioned step 1703.

[0136] It means that the movement magnitude of the direction of X and the direction of Y was computed by the above processing. In addition, YES, the location of the direction of X of a character pattern and the character pattern of a specifications object where it was further created at step 1706 in YES, and the character pattern that it will reach, and the location of the direction of Y will be in agreement, and was created will exist in the optimal location at step 1703.

[0137] Based on the movement magnitude of the direction of X searched for by the arrangement operation part 1060 as mentioned above, and the direction of Y, the created character pattern can be moved to the optimal location (relocation).

[0138] Although this migration processing (relocation) will be performed by the relocation section 1070, since the relocation processing by the relocation section 1070 is the same as that of the procedure shown in drawing 9 of the 1st example, that explanation is omitted here.

[0139]
[Effect of the Invention] As explained above, according to the 1st invention, it is based on the metrics information which shows the physical relationship of the circumscribed quadrangle of the alphanumeric alphabetic character concerned to the zero of an alphanumeric character pattern. Judge whether the location of that alphanumeric alphabetic character is the optimal, and about the alphanumeric alphabetic character which does not exist in the optimal location as a result of this judgment Since it is rearranged according to the movement magnitude calculated based on the above-mentioned metrics information, the character-pattern automatic correction equipment which can correct the location of the alphanumeric character pattern corresponding to an alphanumeric alphabetic character (for example, ASCII code) to the optimal location can

be offered. Therefore, it becomes possible to draw up the document of an exact alphabetic character list.

[0140] Moreover, according to the 2nd invention, it is based on the metrics information which shows the physical relationship of the circumscribed quadrangle of the alphabetic character concerned to the zero of the character pattern containing an alphanumeric character pattern. Judge whether the location of that alphabetic character is the optimal, and about the alphabetic character which does not exist in the optimal location as a result of this judgment Since it is rearranged according to the movement magnitude calculated based on the above-mentioned metrics information and the metrics information about the character pattern of a specifications object The character-pattern automatic correction equipment which can correct the location of the character pattern corresponding to alphabetic characters (for example, character code of JIS), such as an alphanumeric alphabetic character and ** kanas, to the optimal location can be offered. Therefore, it becomes possible to draw up the document of an exact alphabetic character list.

[Translation done.]

Your Ref: 07844-412JP1
Our Ref: PA973

Translation of Selected Portions of Pat. Laid-open Official Gazette

---

Appln. No: 4-174410
Appln. Date: July 1, 1992
Laid-open Pub. No: 6-20026
Laid-open Pub. Date: January 28, 1994

Inventor(s): Masahiko Muramatsu
Applicant(s): Fuji Xerox K.K.
Attorney(s): Takahisa Kimura

---

1. Title of the Invention

CHARACTER PATTERN AUTOMATIC CORRECTING APPARATUS

2. Claims

(omitted)

3. Detailed Description of the Invention (Selected Portions)

1)

(omitted)

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平6－20026

(43)公開日　平成６年(1994)１月28日

(51)Int.Cl.[5]		識別記号	庁内整理番号	Ｆ Ｉ	技術表示箇所
Ｇ０６Ｆ	15/62	３２５　Ｄ	8125－5L		
		４１０　Ｚ	9287－5L		
Ｂ４１Ｂ	23/00				
	27/00				
			8804－2Ｃ	Ｂ４１Ｊ　3/ 12　Ｂ	

審査請求　未請求　請求項の数２(全 16 頁)　最終頁に続く

(21)出願番号　特願平4－174410

(22)出願日　平成４年(1992)７月１日

(71)出願人　000005496
富士ゼロックス株式会社
東京都港区赤坂三丁目３番５号

(72)発明者　村松　正彦
神奈川県川崎市高津区坂戸３丁目２番１号
ＫＳＰ　Ｒ&Ｄ　ビジネスパークビル
富士ゼロックス株式会社内

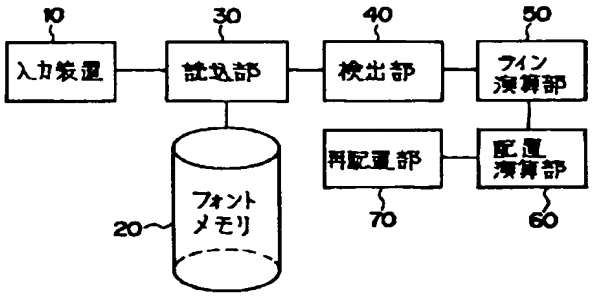
(74)代理人　弁理士　木村　高久

(54)【発明の名称】　文字パターン自動修正装置

(57)【要約】

【目的】英数文字、ひら仮名などの文字パターンの位置を最適な位置に修正することのできる文字パターン自動修正装置を提供する。

【構成】フォントメモリ２０には、文字コードに対応する英数文字パターン及び該英数文字パターンの原点に対する当該英数文字の外接四角形の位置関係を示すメトリクス情報が格納されている。読込部３０は、入力装置１０からの文字コードに対応する英数文字パターン及びそのメトリクス情報をフォントメモリ２０から読み出して、検出手段（検出部４０）に渡す。検出手段は、読込部３０から渡された英数文字パターン及びそのメトリクス情報から、当該英数文字の位置を検出する。演算手段（ライン演算部５０及び配置演算部６０）は、上記メトリクス情報から再配置すべき英数文字の位置を演算する。再配置手段（再配置部７０）は、上記検出手段による検出結果及び上記演算手段による演算結果に基づいて、当該英数文字を再配置する。



1

【特許請求の範囲】
【請求項１】英数文字が表現された英数文字パターンの原点に対する当該英数文字の位置を修正する文字パターン自動修正装置であって、
前記英数文字パターンと、該英数文字パターンの原点に対する当該英数文字の外接四角形の位置関係を示すメトリクス情報とから、当該英数文字の位置を検出する検出手段と、
前記メトリクス情報から再配置すべき前記英数文字の位置を演算する演算手段と、
前記検出手段による検出結果及び前記演算手段による演算結果に基づいて、前記英数文字を再配置する再配置手段とを具えたことを特徴とする文字パターン自動修正装置。
【請求項２】文字が表現された文字パターンの原点に対する当該文字の位置を修正する文字パターン自動修正装置であって、
標準文字が表現された標準文字パターンと、該標準文字パターンの原点に対する当該標準文字の外接四角形の位置関係を示すメトリクス情報を記憶する記憶手段と、
前記文字パターン及び該文字パターンの原点に対する当該文字の外接四角形の位置関係を示すメトリクス情報と、前記記憶手段の記憶内容とに基づいて、当該文字の位置及びその文字に対応する標準文字の位置を検出する検出手段と、
前記検出手段による検出結果に基づいて前記文字の移動量を演算する演算手段と、
前記演算手段による演算結果に基づいて前記文字の再配置を行う再配置手段とを具えたことを特徴とする文字パターン自動修正装置。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】この発明は、文字の位置を自動修正する文字パターン自動修正装置に関する。
【０００２】
【従来の技術】従来においては、ユーザ側で新規に文字パターンを作成しシステムに登録することにより、その文字を使用することが可能になっている。その文字パターンは、ユーザによってベースライン等の位置や文字並びも考慮されて作成されるが、人為的なミスも多い。また文字並びを全く考慮せずに文字パターンが作成されている場合もある。
【０００３】このような問題を解決すべく、文字パターンを検査するものとしては、特開昭６０－２９７８３号公報、特開平３－１７６７６４号公報に開示されたものが知られている。
【０００４】特開昭６０－２９７８３号公報に開示されたものは、作成文字パターンと標準文字パターンとを同一画面上に並べて表示するか、或いは同一紙面上に並べて印刷して、両文字パターンを目視で検査するようにしたものである。

2

【０００５】また特開平３－１７６７６４号公報に開示されたものは、アルファベットの大文字を構成するドットパターンの最下部のライン位置を検出することにより文字画像からベースラインの位置を検出し、この検出した位置と基準値とを比較することにより、アルファベットの大文字についてのベースラインが正しいか否かを判定するようにしたものである。
【０００６】
【発明が解決しようとする課題】しかしながら、上記特開昭６０－２９７８３号公報に開示されたものでは、デザイン的な字体検査のみに限定しているため、ベースラインやＣａｐｈｅｉｇｈｔ、Ｘｈｅｉｇｈｔ等の位置検査には不向きであった。
【０００７】また、上記特開平３－１７６７６４号公報に開示されたものでは、オーバハング（overhang）と呼ばれる視覚補正のためのデザイン的な手法による調整量を考慮していないので、正確な文字パターンを得ることができない。またアルファベットの大文字についての検査は可能であるが、アルファベットの小文字、数字、ひら仮名、カタカナなどの文字についてはチェックすることができない。すなわち、アルファベットの大文字のドットパターンのチェックに限定しているため、ＪＩＳのような７０００文字近くもある文字パターンの位置をチェックすることは不可能である。
【０００８】この発明は、英数文字、ひら仮名などの文字パターンの位置を最適な位置に修正することのできる文字パターン自動修正装置を提供することを目的とする。
【０００９】
【課題を解決するための手段】上記目的を達成するため、第１の発明は、英数文字が表現された英数文字パターンの原点に対する当該英数文字の位置を修正する文字パターン自動修正装置であって、前記英数文字パターンと、該英数文字パターンの原点に対する当該英数文字の外接四角形の位置関係を示すメトリクス情報とから、当該英数文字の位置を検出する検出手段と、前記メトリクス情報から再配置すべき前記英数文字の位置を演算する演算手段と、前記検出手段による検出結果及び前記演算手段による演算結果に基づいて、前記英数文字を再配置する再配置手段とを具えている。
【００１０】また第２の発明は、文字が表現された文字パターンの原点に対する当該文字の位置を修正する文字パターン自動修正装置であって、標準文字が表現された標準文字パターンと、該標準文字パターンの原点に対する当該標準文字の外接四角形の位置関係を示すメトリクス情報を記憶する記憶手段と、前記文字パターン及び該文字パターンの原点に対する当該文字の外接四角形の位置関係を示すメトリクス情報と、前記記憶手段の記憶内容とに基づいて、当該文字の位置及びその文字に対応す

る標準文字の位置を検出する検出手段と、前記検出手段による検出結果に基づいて前記文字の移動量を演算する演算手段と、前記演算手段による演算結果に基づいて前記文字の再配置を行う再配置手段とを具えている。

【００１１】

【作用】第１の発明の文字パターン自動修正装置では、検出手段によって、英数文字パターン及び該英数文字パターンの原点に対する当該英数文字の外接四角形の位置関係を示すメトリクス情報から当該英数文字の位置が検出され、また演算手段によって、前記メトリクス情報から再配置すべき英数文字の位置が演算され、更に再配置手段によって、前記検出手段による検出結果及び前記演算手段による演算結果に基づいて、前記英数文字が再配置される。従って、アルファベットの大文字及び小文字や数字の英数文字パターンの位置を、最適な位置に自動修正することができる。

【００１２】また、第２の発明の文字パターン自動修正装置では、記憶手段には、標準文字が表現された標準文字パターン及び該標準文字パターンの原点に対する当該標準文字の外接四角形の位置関係を示すメトリクス情報が記憶されており、検出手段によって、前記文字パターン及び該文字パターンの原点に対する当該文字の外接四角形の位置関係を示すメトリクス情報と、前記記憶手段の記憶内容とに基づいて、当該文字の位置及びその文字に対応する標準文字の位置が検出され、また演算手段によって、前記検出手段による検出結果に基づいて前記文字の移動量が演算され、更に再配置手段によって、前記演算手段による演算結果に基づいて前記文字が再配置される。従って、アルファベットの大文字及び小文字や数字、ひら仮名文字などのＪＩＳの文字コードに対応する文字パターンの位置を、最適な位置に自動修正することができる。

【００１３】

【実施例】以下、本発明の第１の実施例及び第２の実施例について添付図面を参照して説明する。

【００１４】最初に第１の実施例について、図１乃至図９を参照して説明する。

【００１５】図１は本発明に係る文字パターン自動修正装置の一実施例を機能ブロック図で示したものである。

【００１６】同図において、文字パターン自動修正装置は、文字フォントに対応する文字コードを入力する入力装置１０と、英数文字パターンを格納しているフォントメモリ２０と、指定された文字コードに対応する文字パターンをフォントメモリ２０から読み込む読込部３０と、各文字の位置を検出する検出部４０と、英数文字の並び線（ライン）を求めるライン演算部５０と、検出された位置と移動すべき位置とを求めこれらの誤差（移動量）を演算する配置演算部６０と、配置演算部６０により求められた移動量に基づいて文字を再配置する再配置部７０とを有して構成されている。

【００１７】ここで、フォントメモリ２０には、ＡＳＣＩＩ（アスキー）の文字コードと、その文字コードに対応する英数文字パターン（アウトライン又はビットマップ）及びメトリクス情報からなるパターン情報とが格納されている。

【００１８】このパターン情報について、図２に示す“７”という１つのアウトラインを例にとって説明する。

【００１９】メトリクス情報は、“７”というアウトラインを表現するための各直線の交点である６つの制御点Ｃと、図２中点線で囲まれたバウンディングボックス（文字パターンに接する最大矩形枠）情報と、文字の送り幅であるエスケイプメントＷの情報とを有して構成されている。バウンディングボックス情報は、原点（０，０）からのオフセット値であるアセンダーＡ（正の値）、ディセンダーＤ（正の値）、ライトベアリングＲ（正の値）及びレフトベアリングＬ（負の値）の４つの情報から構成されている。

【００２０】また英数文字には原則的な並び線（ライン）が幾つかあって、各文字はそれらの線に沿って並んでいるのが普通である。それらの各ラインについて図３を参照して説明する。

【００２１】この実施例では、８つのラインＬＮ１、ＬＮ２、ＬＮ３、ＬＮ４、ＬＮ５、ＬＮ６、ＬＮ７、ＬＮ８が設定されている。

【００２２】ＬＮ１はアセンダーライン（ascender line ）を示し、ＬＮ２はキャップライン（cap line）を示し、ＬＮ４はミーンライン（mean line ）を示し、ＬＮ５はベースライン（base line ）を示し、ＬＮ７はディセンダーライン（decender line ）を示している。ここでアセンダーラインＬＮ１とはアルファベットの小文字の最も高いラインのことであり、ディセンダーラインＬＮ７とはアルファベットの小文字の最も低いラインのことである。

【００２３】ラインＬＮ３、ＬＮ６、ＬＮ８はオーバハング（overhang）を求めるために必要なラインである。ここでオーバハングとは視覚補正のためのデザイン的な手法による調整のことである。このオーバハングには、キャップ・オーバハング（cap overhang）、ミーン・オーバハング（mean overhang ）、ベース・オーバハング（base overhang ）及びディセンダー・オーバハング（decender overhang ）の４種類があり、これらは次のようにして求めることができる。

【００２４】

キャップ・オーバハング＝ラインＬＮ１の値－ラインＬＮ２の値　…（１）

ミーン・オーバハング＝ＬＮ３の値－ＬＮ４の値　…（２）

ベース・オーバハング＝ＬＮ５の値－ＬＮ６の値　…（３）

ディセンダー・オーバハング＝ＬＮ７の値－ＬＮ８の値　　…（４）

これらのオーバハングはライン演算部５０によって演算される。

【００２５】この実施例においては、ベース・ラインＬＮ５の値はＹ座標値＝０としているので、アセンダー・ラインＬＮ１、キャップ・ラインＬＮ２、ラインＬＮ３、ミーン・ラインＬＮ４、及びベース・ラインＬＮ５の各値は正の値となり、これに対し、ラインＬＮ６、ディセンダー・ラインＬＮ７及びラインＬＮ８の各値は負の値となる。

【００２６】なおこの実施例では、上述した検出手段は検出部４０を有して構成されており、上述した演算手段はライン演算部５０及び配置演算部６０を有して構成されており、上述した再配置手段は再配置部７０を有して構成されている。

【００２７】次に具体例を挙げて文字パターンの位置修正を説明する。

【００２８】入力装置１０から、文字“Ａ”、“ｎ”、“ｉ”、“Ｏ”の各文字にそれぞれ対応する文字コードが順次入力されると、読出部２０は、フォントメモリ２０からフォントファイルを読み込み、そのファイル内から、それらの文字コードに対応する文字パターンを読み込む。この結果、図４に示すような英数文字パターンの並びが得られることとなる。しかし、これらの英数文字パターンは、図４に示すように最適な並びでないということが分かる。なおこの実施例では、前提条件として英数文字パターンのデザインは正しいものとする。

【００２９】そこで、それらの文字パターンの位置を検査し、最適な位置に修正する必要があるので、次に、本発明に係る文字パターンの位置の検査及び修正について説明する。

【００３０】ここでは、最初の文字“Ａ”を例にとって説明することにし、その文字の位置を検査し修正する処理過程を図５を用いて説明する。

【００３１】読込部３０でフォントファイルを読み込み、文字コードによって文字を判別する。この場合、ＡＳＣＩＩコードになっているため大文字の“Ａ”はＯｘ４１である。すなわち、読込部３０は、入力装置１０からの文字コードＯｘ４１を受け取ると、大文字の“Ａ”であると判定する。

【００３２】ここで、大文字の“Ａ”のパターンは図５（ａ）に示すような内容であり、そのバウンディングボックスは、エスケイプメントＷの中心より左側（紙面に向かって左側）に存在し、かつバウンディングボックスの底辺がディセンダーラインに接した位置に存在しているものとする。

【００３３】検出部４０では、大文字“Ａ”のバウンディングボックス情報から、その重心位置を検出する。ここでは、バウンディングボックスの対角線の交点を重心位置とする。具体的には、バウンディングボックスは、ディセンダー・ラインにおけるライトベアリングの値及びレフトベアリングの値でそれぞれ示される点、及びアセンダー・ラインにおけるライトベアリングの値及びレフトベアリングの値でそれぞれ示される点を頂点とする矩形領域で表現することができる。従って、これらの値に基づいて対角線の交点を求めることができる。

【００３４】ライン演算部５０では、図３に示した様な各ラインを求める（この演算の詳細については後述する）。

【００３５】配置演算部６０では、重心位置を示す座標値（Ｘ０、Ｙ０）がエスケイプメントＷの中心に存在するか否かを調べる。中心に存在しなければ、Ｘ方向の移動量を計算する。この場合、移動量ＸＭ＝Ｗ－Ｘ０となる。

【００３６】この様にＸ方向の移動量を演算したならば、配置演算部６０では、次にＹ方向の移動量を計算する。“Ａ”の場合には、図３に示したベースラインＬＮ５がバウンディングボックスの底辺と接した状態が適切な位置であるので、この文字に対するディセンダーの値は「０」である。しかし、この例では、バウンディングボックスの底辺がディセンダーラインに接しているので、この場合のＹ方向の移動量はディセンダーＤの値である。

【００３７】再配置部７０では、Ｘ方向にＸＭ、Ｙ方向にｄだけ移動する。これにより、バウンディングボックスは、図５（ｂ）に示すようにＸ方向にＸＭだけ移動され、さらに、図５（ｂ）に示すようにＹ方向にディセンダーＤの値だけ移動される。なお移動量に基づくＸ方向、Ｙ方向への移動の際はその順序は問わない。

【００３８】上述したように大文字“Ａ”のようにバウンディングボックスの底辺がベースラインに接する他の文字については同様の処理で文字の位置を修正することができる。

【００３９】このような文字以外の文字についても、Ｘ方向の移動量については上記同様の方法で求めることができ、一方Ｙ方向の移動量については、バウンディングボックスの底辺がラインＬＮ５、ＬＮ６、ＬＮ７、ＬＮ８のいずれかと接するのかが分かれば求めることができる。

【００４０】一般的に、アルファベット｛Ａ、Ｂ、Ｄ、Ｅ、Ｆ、Ｈ、Ｉ、Ｋ、Ｌ、Ｍ、Ｎ、Ｐ、Ｒ、Ｔ、Ｖ、Ｗ、Ｘ、Ｙ、Ｚ｝及び数字｛１、２、４、７｝は共に、バウンディングボックスの底辺が図３に示したベースラインＬＮ５と接する。

【００４１】またアルファベット｛Ｃ、Ｇ、Ｏ、Ｑ、Ｓ、Ｕ｝及び数字｛０、３、６、８、９｝は共に、バウンディングボックスの底辺が図３に示したラインＬＮ６と接する。

【００４２】これに対し、アルファベットの小文字は、

高さとラインの並びによってショートレター、アセンダーレター、ディセンダーレター、ロングレターの４種類に分類することができるので、その種類に応じた配置となる。

【００４３】ショートレターのうち、｛ａ、ｃ、ｅ、ｏ、ｓ、ｕ｝は共にバウンディングボックスの底辺が図３に示したラインＬＮ６と接し、また｛ｍ、ｎ、ｒ、ｖ、ｗ、ｘ、ｚ｝は共にバウンディングボックスの底辺が図３に示したベースラインＬＮ５と接する。

【００４４】アセンダーレターのうち、｛ｂ、ｄ、ｔ｝は共にバウンディングボックスの底辺が図３に示したラインＬＮ６と接し、また｛ｈ、ｉ、ｋ、ｌ｝は共にバウンディングボックスの底辺が図３に示したベースラインＬＮ５と接する。

【００４５】ディセンダーレターのうち、｛ｐ、ｑ｝は共にバウンディングボックスの底辺が図３に示したディセンダーラインＬＮ７と接し、また｛ｇ、ｙ｝は共にバウンディングボックスの底辺が図３に示したラインＬＮ８と接する。

【００４６】ロングレター｛ｊ｝はバウンディングボックスの底辺が図３に示したラインＬＮ８と接する。

【００４７】この実施例においては、アルファベットの小文字は上述した規則に従うものとする。

【００４８】次にライン演算部５０によるベースライン以外の各ラインの求め方について、図６及び図７を参照して説明する。

【００４９】ここで前提条件として、入力するフォントには文字“Ｈ”、“Ｏ”、“ｘ”、“ｐ”が必ずあるものとする。なお、入力する文字はこれらの文字に限定されるものではなく、文字“Ｈ”と同様にバウンディングボックスの底辺がベースラインＬＮ５と接する文字、文字“Ｏ”と同様にバウンディングボックスの底辺がラインＬＮ６と接する文字、ショートレータ“ｘ”と同様にバウンディングボックスの底辺がベースラインＬＮ５と接する文字、ディセンダーレター“ｐ”と同様にバウンディングボックスの底辺がディセンダーラインＬＮ７と接する文字にそれぞれ対応する文字を入力するようにしても良い。

【００５０】なお図６及び図７において、“ＢＢｏｘ”はバウンディングボックスを示し、“overhang”はオーバハングを示し、“cap overhang”はキャップ・オーバハングを示し、“mean overhang ”はミーン・オーバハングを示し、“base overhang ”はベース・オーバハングを示し、“descender overhang”はディセンダー・オーバハングを示している。

【００５１】最初に、overhang（cap overhang＝mean overhang ＝base overhang ＝descender overhangと仮定する）を求める。この演算処理について、図６に示すフローチャートを参照して説明する。

【００５２】ライン演算部５０は、文字コードを読み込んで（ステップ６０１）、大文字の“Ｈ”の文字コードか否かを判断する（ステップ６０２）。大文字の“Ｈ”の文字コードである場合は、大文字の“Ｈ”のメトリクス情報を読み込んで（ステップ６０３）、その文字のバウンディングボックスＢＢｏｘの高さを求める（ステップ６０４）。

【００５３】そのバウンディングボックスＢＢｏｘの高さはアセンダーの値とディセンダーの値とを加算することにより得られる。このアセンダーの値及びディセンダーの値は、今読み込んだ大文字の“Ｈ”のメトリクス情報に含まれているので、そのメトリクス情報から容易に知ることができる。

【００５４】また上記ステップ６０２において大文字の“Ｈ”の文字コードでない場合は、次に大文字の“Ｏ”の文字コードか否かを判断する（ステップ６０５）。

【００５５】大文字の“Ｏ”の文字コードである場合は、大文字の“Ｏ”のメトリクス情報を読み込んで（ステップ６０６）、その文字のバウンディングボックスＢＢｏｘの高さを求める（ステップ６０７）。この場合のバウンディングボックスＢＢｏｘの高さも上記同様にして求めることができる。

【００５６】このようにして大文字の“Ｈ”及び“Ｏ”のバウンディングボックスＢＢｏｘの高さを求めたならば、ライン演算部５０は、次式を演算することによりオーバハングを求める（ステップ６０８）。

【００５７】

$$\text{overhang} = (\text{“O”のBBoxの高さ} - \text{“H”のBBoxの高さ}) / 2 \quad \cdots (5)$$

この式（５）を演算することにより求められたオーバハングの値がキャップ・オーバハング及びベース・オーバハングとなる。

【００５８】上記ステップ６０５において大文字の“Ｏ”の文字コードでない場合は、小文字の“ｘ”の文字コードか否かを判断する（ステップ６０９）。

【００５９】小文字の“ｘ”の文字コードである場合は、小文字の“ｘ”のメトリクス情報を読み込んで（ステップ６１０）、上記同様にして、その文字のバウンディングボックスＢＢｏｘの高さを求める（ステップ６１１）。

【００６０】一方、ステップ６０９において小文字の“ｘ”の文字コードでない場合は、小文字の“ｐ”の文字コードか否かを判断する（ステップ６１２）。

【００６１】小文字の“ｐ”の文字コードの場合は、小文字の“ｐ”のメトリクス情報を読み込んみ（ステップ６１３）、その後、ステップ６１１に進む。

【００６２】なおステップ６１２において小文字の“ｐ”の文字コードでない場合はステップ６０１に戻りこのステップ以降を実行する。

【００６３】次にライン演算部５０による上記各ラインの決定処理について、図７に示すフローチャートを参照して説明する。

【００６４】最初に、ライン演算部５０は、cap overhang＝mean overhang ＝base overhang ＝descender overhangと定義し（ステップ７０１）、この定義内容と、予め設定されたベース・ラインの値とに基づいて、ラインＬＮ１、ＬＮ２、ＬＮ３、ＬＮ４、ＬＮ６、ＬＮ７、ＬＮ８を求める。具体的には次の様にして各ラインを求めるようにしている。

【００６５】最初に、ベース・ラインＬＮ５の値からベース・オーバハングの値を減算することにより、ラインＬＮ６を求めることができる（ラインＬＮ６＝ベース・ラインＬＮ５の値－ベース・オーバハングの値）（ステップ７０２）。このラインの値は負の値となる。

【００６６】次にベース・ラインＬＮ５の値と“Ｈ”のＢＢｏｘの高さとを加算することにより、キャップ・ラインＬＮ２を求めることができる（キャップ・ラインＬＮ２＝ベース・ラインＬＮ５の値＋“Ｈ”のＢＢｏｘの高さ）（ステップ７０３）。このラインの値は正の値となる。

【００６７】こうして得られたキャップ・ラインＬＮ２の値とキャップ・オーバハングの値とを加算することにより、アセンダー・ラインＬＮ１を求めることができる（アセンダー・ラインＬＮ１＝キャップ・ラインＬＮ２の値＋キャップ・オーバハングの値）（ステップ７０４）。このラインの値は正の値となる。

【００６８】ベース・ラインＬＮ５の値と小文字“ｘ”のＢＢｏｘの高さと加算することにより、ミーン・ラインＬＮ４を求めることができる（ミーン・ラインＬＮ４＝ベース・ラインＬＮ５の値＋“ｘ”のＢＢｏｘの高さ）（ステップ７０５）。このラインの値は正の値となる。

【００６９】こうして求められたミーン・ラインＬＮ４の値とミーン・オーバハングの値とを加算することにより、ラインＬＮ３を求めることができる（ラインＬＮ３＝ミーン・ラインＬＮ４の値＋ミーン・オーバハングの値）（ステップ７０６）。このラインの値は正の値となる。

【００７０】また、こうして求められたＬＮ３の値から、小文字“ｐ”のＢＢｏｘの高さを減算することにより、ディセンダー・ラインＬＮ７を求めることができる（ディセンダー・ラインＬＮ７＝ラインＬＮ３の値－“ｐ”のＢＢｏｘの高さ）（ステップ７０７）。このラインの値は負の値となる。

【００７１】さらに、こうして求められたディセンダー・ラインＬＮ７の値から、ディセンダー・オーバハングの値を減算することにより、ラインＬＮ８を求めることができる（ラインＬＮ８＝ディセンダー・ラインＬＮ７の値－ディセンダー・オーバハングの値）（ステップ７０８）。このラインの値は負の値となる。

【００７２】以上の処理でライン演算部５０により各ラインが決定されたので、それらの各ラインの値に基づいて、各文字のＹ方向の移動量を演算することができる。

【００７３】次に、配置演算部６０による各文字のＹ方向の移動量の演算処理について、図８に示すフローチャートを参照して説明する。

【００７４】配置演算部６０は、文字コードを読み込み（ステップ８０１）、次に示す文字群中のいずれかの文字に対応する文字コードであるか否かを判断する（ステップ８０２）。

【００７５】すなわち、アルファベットの大文字｛Ａ、Ｂ、Ｄ、Ｅ、Ｆ、Ｈ、Ｉ、Ｋ、Ｌ、Ｍ、Ｎ、Ｐ、Ｒ、Ｔ、Ｖ、Ｗ、Ｘ、Ｙ、Ｚ｝、数字｛１、２、４、７｝、アルファベットの小文字｛ｍ、ｎ、ｒ、ｖ、ｗ、ｘ、ｚ、ｈ、ｉ、ｋ、ｌ｝のいずれかの文字の文字コードであるか否かを判断する。

【００７６】ここで、いずれかの文字コードである場合は、その文字のバウンディングボックスＢＢｏｘの底辺がベース・ラインＬＮ５に接するように、Ｙ方向への移動量を求める（ステップ８０３）。

【００７７】ステップ８０３を終了した場合、ステップ８０２において該当する文字の文字コードでない場合は、次に示す文字群中のいずれかの文字に対応する文字コードであるか否かを判断する（ステップ８０４）。

【００７８】すなわち、アルファベットの大文字｛Ｃ、Ｇ、Ｏ、Ｑ、Ｓ、Ｕ｝、数字｛０、３、６、８、９｝及びアルファベットの小文字｛ａ、ｃ、ｅ、ｏ、ｓ、ｕ、ｂ、ｄ、ｔ｝のいずれかの文字の文字コードであるか否かを判断する。

【００７９】ここで、いずれかの文字コードである場合は、その文字のバウンディングボックスＢＢｏｘの底辺がラインＬＮ６に接するように、Ｙ方向への移動量を求める（ステップ８０５）。

【００８０】ステップ８０５を終了した場合、ステップ８０４において該当する文字の文字コードでない場合は、次に示す文字群中のいずれかの文字に対応する文字コードであるか否かを判断する（ステップ８０６）。

【００８１】すなわち、アルファベットの小文字｛ｐ、ｑ｝のいずれかの文字の文字コードであるか否かを判断する。

【００８２】ここで、いずれかの文字コードである場合は、その文字のバウンディングボックスＢＢｏｘの底辺がディセンダー・ラインＬＮ７に接するように、Ｙ方向への移動量を求める（ステップ８０７）。

【００８３】ステップ８０７を終了した場合、ステップ８０６において該当する文字の文字コードでない場合は、次に示す文字群中のいずれかの文字に対応する文字コードであるか否かを判断する（ステップ８０８）。

【００８４】すなわち、アルファベットの小文字｛ｇ、

ｙ、ｊ｝のいずれかの文字の文字コードであるか否かを判断する。

【００８５】ここで、いずれかの文字コードである場合は、その文字のバウンディングボックスＢＢｏｘの底辺がラインＬＮ８に接するように、Ｙ方向への移動量を求める（ステップ８０９）。

【００８６】ステップ８０９を終了した場合、ステップ８０８において該当する文字の文字コードでない場合は、処理すべき文字は存在するか否かを判断し（ステップ８１０）、存在する場合には上記ステップ８０１に戻りこのステップ以降を実行し、一方、存在しない場合は処理を終了する。

【００８７】こうして求められた移動量に基づいて文字を最適な位置に再配置することができる。

【００８８】次に、再配置部７０による文字の再配置処理について、図９に示すフローチャートを参照して説明する。

【００８９】再配置部７０は、キャラクタのＸ方向及びＹ方向の移動量として座標値を求める。ここで、Ｘ方向の移動量として検出部４０によって求められた移動量とし、これをＸｍ１と定義し、またＹ方向の移動量として配置演算部６０によって求められた移動量とし、これをＹｍ２と定義する（ステップ９０１）。

【００９０】移動量（Ｘｍ１、Ｙｍ２）を求めたら、バウンディングボックスＢＢｏｘに関する各値を変更する。

【００９１】すなわち、アセンダー及びディセンダーについてはそれぞれＹ方向にＹｍ２だけ移動し、レフトベアリング及びライトベアリングについてはそれぞれＸ方向にＸｍ１だけ移動する（ステップ９０２）。

【００９２】次に、再配置部７０は、対象の文字パターンがビットマップで表現されているか否かを判断する（ステップ９０３）。

【００９３】ビットマップであれば、ドットの存在する座標点（ｘｉ、ｙｉ）を、Ｘｍ１、Ｙｍ２だけ移動する。つまり、ｎ個のドットの座標点を（ｘｉ、ｙｉ）（ｉ＝１、２、…ｎ（整数））とすれば、ｘｉ＝ｘｉ＋Ｘｍ１、ｙｉ＝ｙｉ＋Ｙｍ２の各式を演算して得られる座標にドットを移動させる（ステップ９０４）。

【００９４】ステップ９０３においてビットマップでない場合（この場合はアウトラインということになる）は、ｎ個の制御点の座標点を（ｘｉ、ｙｉ）（ｉ＝１、２、…ｎ（整数））とすれば、ｘｉ＝ｘｉ＋Ｘｍ１、ｙｉ＝ｙｉ＋Ｙｍ２の各式を演算して得られる座標にドットを移動させる（ステップ９０５）。

【００９５】ステップ９０４を終了した場合、ステップ９０５を終了した場合は、処理すべき文字が存在するか否かを判断する（ステップ９０６）。

【００９６】ここで、存在する場合には上記ステップ９０１に戻りこのステップ以降を実行し、一方、存在しない場合は処理を終了する。

【００９７】以上の処理により、英数文字パターンの位置が最適な位置に自動修正されたこととなる。

【００９８】次に本発明の第２の実施例を、図１０乃至図１７を参照して説明する。

【００９９】図１０は本発明に係る文字パターン自動修正装置の第２の実施例を機能ブロック図で示したものである。

【０１００】同図において、文字パターン自動修正装置は、文字フォントに対応する文字コードを入力する入力装置１０１０と、作成された文字パターンを格納しているフォントメモリ１０２０と、指定された文字コードに対応する文字パターンをフォントメモリ１０２０から読み込む読込部１０３０と、予め設定された標準書体の文字パターンを格納している標準書体フォントメモリ１０４０と、読込部１０３０によって読み込まれた文字パターンの位置を検出すると共に、標準書体フォントメモリ１０４０からその文字パターンの文字コードと対応する標準書体の文字パターンを読み込んでこの文字パータンの位置を検出する検出部１０５０と、作成された文字パターンの位置と標準書体の文字パターンの位置とに基づいて作成された文字の移動すべき位置を演算する配置演算部１０６０と、配置演算部１０６０により求められた位置に文字を再配置する再配置部１０７０とを有して構成されている。

【０１０１】フォントメモリ２０には、ＪＩＳの文字コードと、その文字コードに対応する文字パターン（アウトライン又はビットマップ）及びメトリクス情報からなるパターン情報とが格納されている。ここには、例えば図１１（ａ）、（ｂ）に示すような作成された文字パターンが格納される。ここでは、前提条件として文字のデザインは正しいものとする。

【０１０２】標準書体フォントメモリ１０４０には、あらゆる書体の模範となる標準書体の文字パターンが格納されており、具体的には、上記同様に、ＪＩＳの文字コードと、その文字コードに対応する文字パターン（アウトライン又はビットマップ）及びメトリクス情報からなるパターン情報とが格納されている。ここには、例えば例えば図１２に示すような標準書体の文字パターンが格納されている。ここでは、例えばＪＩＳ　Ｘ０２０８の６８７７文字が格納されているものとし、また前提条件として標準書体のデザイン、文字の並びともに正しいデータであるものとする。

【０１０３】ここで、上記パターン情報について、図１３に示す“Ａ”を例にとって説明する。

【０１０４】メトリクス情報は、“Ａ”という文字マスクと、図１３中点線で囲まれたバウンディングボックスＢＢｏｘ（文字パターンに接する最大矩形枠）の情報と、文字の送り幅であるエスケイプメントＷの情報とを有して構成されている。バウンディングボックスＢＢｏ

ｘ情報は、原点（０，０）からのオフセット値であるアセンダーＡ（正の値）、ディセンダーＤ（負の値）、ライトベアリングＲ（正の値）及びレフトベアリングＬ（負の値）の４つの情報から構成されている。このようなメトリクス情報は、作成された文字パターン及び標準書体の文字パターンそれぞれについて定義される。

【０１０５】なお、この実施例では、上述した記憶手段は標準書体フォントメモリ１０４０を有して構成されており、上述した検出手段は検出部１０５０を有して構成されており、上述した演算手段は配置演算部１０６０を有して構成されており、再配置手段は再配置部１０７０を有して構成されている。

【０１０６】係る構成において、文字パターン自動修正装置の文字パターンの位置修正処理について説明する。

【０１０７】入力装置１０１０から、文字“Ａ”、“Ｂ”、“あ”、“い”の各文字にそれぞれ対応する文字コードが順次入力されると、読出部１０３０は、フォントメモリ１０２０からフォントファイルを読み込み、そのファイル内から、それらの文字コードに対応する文字パターンを読み込む。この結果、図１１（ａ）に示すような文字パターンの並びが得られることとなる。しかし、これらの文字パターンは、図１２に示すように最適な並びでないということが分かる。

【０１０８】そこで、それらの文字パターンの位置を検査し、最適な位置に修正する必要があるので、次に、本発明に係る文字パターンの位置の検査及び修正について説明する。

【０１０９】ここでは、最初の文字“Ａ”を例にとって説明することにし、その文字の位置を検査し修正する処理過程を図１４乃至図１６を参照して説明する。

【０１１０】読込部１０３０でフォントファイルを読み込み、文字コードによって文字を判別する。この場合、ＪＩＳコードになっているため大文字の“Ａ”はＯｘ２３４１（ＪＩＳ３区３３点）である。すなわち、読込部１０３０は、入力装置１０１０からの文字コードＯｘ２３４１を受け取ると、該文字コードに基づいて大文字の“Ａ”であると判定し、更に、その文字の文字パターン及びメトリクス情報をフォントファイルから読み取ると共に、読み取ったこれらの情報及び文字コードを検出部１０５０に渡す。

【０１１１】検出部１０５０では、文字“Ａ”のバウンディングボックス情報（アセンダー、ディセンダー、レフトベアリング、ライトベアリング）から、図１４に示されるｙ０、ｙ１、ｘ０、ｘ１を検出する。図１４において、ＭＸは文字が登録される予め設定された領域（ボデーサイズ）の縦の長さ、ＭＹは前記領域（ボデーサイズ）の横の長さを示している。

【０１１２】すなわち、
ｙ０はディセンダーの値の絶対値と同一である。
（ｙ０＝－（ディセンダーの値））
ｙ１はＭＹの値からアセンダーの値を減算して得られる値である。
（ｙ１＝ＭＹの値－アセンダーの値）
ｘ０はレフトベアリングの値の絶対値と同一てある。
（ｘ０＝－（レフトベアリングの値））
ｘ１はＭＸの値からライトベアリングの値を減算して得られる値である。
（ｘ１＝ＭＸの値－ライトベアリングの値）
このようにして求められたｙ０、ｙ１、ｘ０、ｘ１の値は作成された文字パターンに関する位置情報として定義される。

【０１１３】次に検出部１０５０は、標準書体の文字“Ａ”、すなわち文字コードＯｘ２３４１の文字を標準書体フォントメモリ１０４０から読み出す。ここで、標準書体の文字“Ａ”は図１５に示すような内容になっているものとする。図１５において、ＭＸ０はボデーサイズの縦の長さ、ＭＹ０はボデーサイズの横の長さを示している。この標準文字パターンについても、標準書体の文字“Ａ”のバウンディングボックス情報から、図１５に示されるｙａ、ｙｂ、ｘａ、ｘｂを検出する。これらの値は上記同様にして求めることができる。

【０１１４】すなわち、
ｙａ＝ＭＹ０の値－アセンダーの値
ｙｂ＝－（ディセンダーの値）
ｘａ＝－（レフトベアリングの値）
ｘｂ＝ＭＸ０の値－ライトベアリングの値
このようにして求められたｙａ、ｙｂ、ｘａ、ｘｂの値は標準書体の文字パターンに関する位置情報として定義される。

【０１１５】作成された文字パターン及び標準書体の文字パターンに関する位置情報は、検出部１０５０から配置演算部１０６０に出力される。

【０１１６】配置演算部１０６０によって、作成された文字パターンに関する位置情報と標準書体の文字パターンに関する位置情報とを比較し、この比較結果に基づいて最適な位置への移動量を演算し、更に再配置部１０７０によって、その移動量に基づいて、作成された文字パターンを移動させることにより、結果として図１６に示すような文字パターンが得られることとなる。

【０１１７】次に、配置演算部１０６０による移動量の演算処理について、図１７に示すフローチャートを参照して説明する。

【０１１８】配置演算部１０６０は、検出部１０５０からの作成された文字パターンに関する位置情報（ｙ０、ｙ１、ｘ０、ｘ１）を読み込むと共に、Ａ＝（ｘ０＋ｘ１）と定義し、またＢ＝（ｙ０、ｙ１）と定義する（ステップ１７０１）。また検出部１０５０からの標準書体の文字パターンに関する位置情報（ｙａ、ｙｂ、ｘａ、ｘｂ）を読み込む（ステップ１７０２）。

【０１１９】次に、配置演算部１０６０は、ｘ０：ｘ１

＝ｘａ：ｘｂの式が成立するか否かを調べ（ステップ１７０３）、この式が成立する場合はＸ方向の位置は正しいことになるので、次にＹ方向についての移動量を求めることになる。

【０１２０】一方ステップ１７０３において上記式が不成立の場合は、Ｘ方向の文字の位置は最適ではないので、その文字を最適な位置に移動すべく移動量を算出する（ステップ１７０４）。

ｘ０ｎｅｗ：ｘ１ｎｅｗ＝ｘａ：ｘｂ　　…（６）

の式が成立するように、ｘ０ｎｅｗ及びｘ１ｎｅｗの各値を算出する。

【０１２１】すなわち、

ｎｅｗ＝ＭＸの値－（ライトベアリングの値＋レフトベアリングの値）＝Ａ（定数）の関係から、

【０１２２】この場合、ｘ０＋ｘ１＝ｘ０ｎｅｗ＋ｘ１

ｘ１ｎｅｗ＝Ａ－ｘ０ｎｅｗ　　…（７）

の式が得られる。

【０１２３】また、この式（７）の関係から、上記式

ｘ０ｎｅｗ：（Ａ－ｘ０ｎｅｗ）＝ｘａ：ｘｂ　　…（８）

の式に変形することができる。

【０１２４】更に式（８）においてＡ、ｘａ、ｘｂは定数であるので、式（８）からは、

ｘ０ｎｅｗ＝（Ａ×ｘａ）／（ｘａ＋ｘｂ）　　…（９）

の式が得られる。

【０１２５】この式（９）を演算することによりｘ０ｎｅｗの値すなわちＸ方向の最適な位置が得られる。

【０１２６】そして配置演算部１０６０は、上記式（９）を演算して得られたｘ０ｎｅｗの値からｘ０を減算した値をＸ方向の移動量とする（Ｘ方向の移動量＝ｘ０ｎｅｗ－ｘ０）（ステップ１７０５）。

【０１２７】またステップ１７０５を終了した場合、上記ステップ１７０３において式が成立する場合は、配置演算部１０６０は、ｙ０：ｙ１＝ｙｂ：ｙａの式が成立するか否かを調べ（ステップ１７０６）、この式が成立する場合はＹ方向の位置は正しいことになる。

【０１２８】一方ステップ１７０６において上記式が不成立の場合は、Ｙ方向の文字の位置は最適ではないので、その文字を最適な位置に移動すべく移動量を算出する（ステップ１７０７）。

【０１２９】すなわち、

ｙ０ｎｅｗ：ｙ１ｎｅｗ＝ｙｂ：ｙａ　　…（１０）

の式が成立するように、ｙ０ｎｅｗ及びｙ１ｎｅｗの各値を算出する。

【０１３０】この場合、ｙ０＋ｙ１＝ｙ０ｎｅｗ＋ｙ１

ｎｅｗ＝ＭＹの値－（アセンダーの値＋ディセンダーの値）＝Ｂ（定数）の関係から、

ｙ１ｎｅｗ＝Ｂ－ｙ０ｎｅｗ　　…（１１）

の式が得られる。

【０１３１】また、この式（１１）の関係から、上記式

（１０）は、

ｙ０ｎｅｗ：（Ｂ－ｙ０ｎｅｗ）＝ｙｂ：ｙａ　　…（１２）

の式に変形することができる。

【０１３２】さらに式（１２）においてＢ、ｙａ、ｙｂは定数であるので、式（１２）からは、

ｙ０ｎｅｗ＝（Ｂ×ｙｂ）／（ｙａ＋ｙｂ）　　…（１２）

の式が得られる。

【０１３３】この式（１２）を演算することによりｙ０ｎｅｗの値すなわちＹ方向の最適な位置が得られる。

【０１３４】そして配置演算部１０６０は、上記式（１２）を演算して得られたｙ０ｎｅｗの値からｙ０を減算した値をＹ方向の移動量とする（Ｙ方向の移動量＝ｙ０ｎｅｗ－ｘ０）（ステップ１７０８）。

【０１３５】ステップ１７０８を終了した場合、上記ステップ１７０３において式が成立した場合は処理を終了する。

【０１３６】以上の処理で、Ｘ方向及びＹ方向の移動量が算出されたことになる。なお、ステップ１７０３でＹＥＳ、更にステップ１７０６でＹＥＳの場合は、作成された文字パターンと標準書体の文字パターンのＸ方向の位置及び及びＹ方向の位置が一致していることとなり、作成された文字パターンは最適な位置に存在していることになる。

【０１３７】上述したように配置演算部１０６０によって求められたＸ方向及びＹ方向の移動量に基づいて、作成された文字パターンを最適な位置に移動（再配置）させることができる。

【０１３８】この移動処理（再配置）は再配置部１０７０によって行われることになるが、再配置部１０７０による再配置処理は、第１の実施例の図９に示される処理手順と同様なので、ここではその説明を省略する。

【０１３９】

【発明の効果】以上説明したように第１の発明によれば、英数文字パターンの原点に対する当該英数文字の外

接四角形の位置関係を示すメトリクス情報に基づいて、その英数文字の位置が最適か否かを判定し、この判定の結果、最適な位置に存在していない英数文字については、上記メトリクス情報に基づいて演算した移動量に従って再配置されるので、英数文字（例えばアスキーコード）に対応する英数文字パターンの位置を最適な位置に修正することのできる文字パターン自動修正装置を提供することができる。従って、正確な文字並びの文書を作成することが可能となる。

【０１４０】また第２の発明によれば、英数文字パターンを含む文字パターンの原点に対する当該文字の外接四角形の位置関係を示すメトリクス情報に基づいて、その文字の位置が最適か否かを判定し、この判定の結果、最適な位置に存在していない文字については、上記メトリクス情報と標準書体の文字パターンに関するメトリクス情報とに基づいて演算した移動量に従って再配置されるので、英数文字、ひら仮名などの文字（例えばＪＩＳの文字コード）に対応する文字パターンの位置を最適な位置に修正することのできる文字パターン自動修正装置を提供することができる。従って、正確な文字並びの文書を作成することが可能となる。

【図面の簡単な説明】

【図１】本発明に係る文字パターン自動修正装置の第１の実施例を示す機能ブロック図。

【図２】英数文字パターンのメトリクス情報を説明するための図。

【図３】標準的な英数文字パターンの並びを説明するための図。

【図４】修正前の英数文字パターンの並びを説明するための図。

【図５】修正前の英数文字パターンの位置を最適な位置に修正するための処理過程を説明するための図。

【図６】ライン演算部によるオーバハングの算出処理の処理手順を示すフローチャート。

【図７】ライン演算部による英数文字の並び線（ライン）の決定処理の処理手順を示すフローチャート。

【図８】配置演算部による文字パターンの移動量算出処理の処理手順を示すフローチャート。

【図９】再配置部による文字パターンの再配置処理の処理手順を示すフローチャート。

【図１０】本発明に係る文字パターン自動修正装置の第２の実施例を示す機能ブロック図。

【図１１】フォントメモリに格納される作成された文字パターンの一例を示す図。

【図１２】標準書体フォントメモリに格納される標準書体の文字パターンの一例を示す図。

【図１３】文字パターンのメトリクス情報を説明するための図。

【図１４】修正前の作成された文字パターンに関する位置情報の算出を説明するための図。

【図１５】標準書体の文字パターンに関する位置情報の算出を説明するための図。

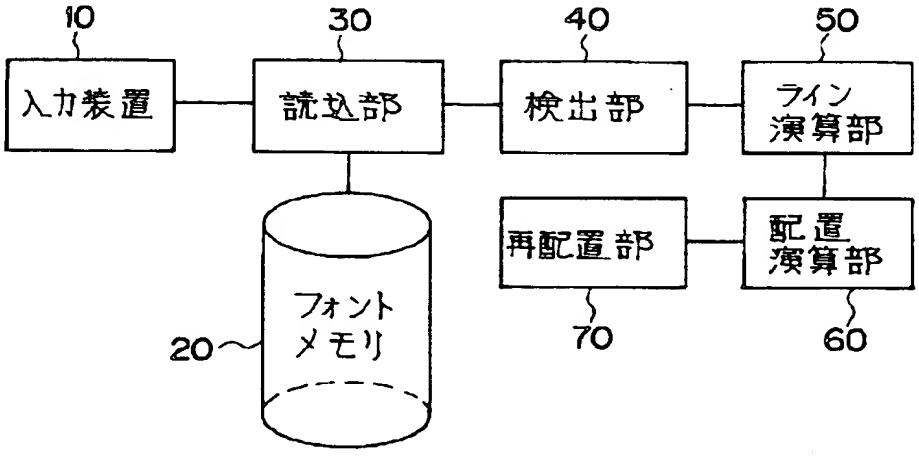
【図１６】修正後の作成された文字パターンに関する位置情報の算出を説明するための図。

【図１７】配置演算部による文字パターンの移動量算出処理の処理手順を示すフローチャート。

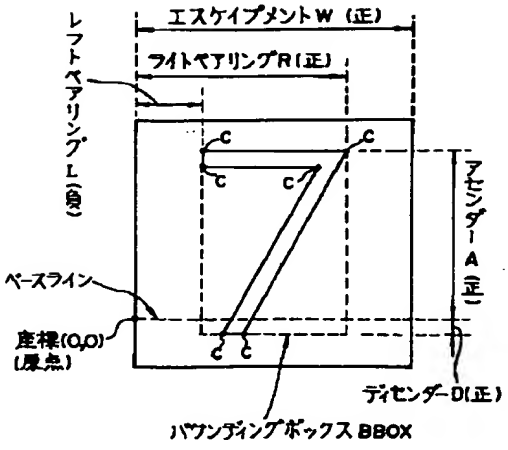
【符号の説明】

１０、１０１０…入力装置、２０、１０２０…フォントメモリ、３０、１０３０…読込部、４０、１０５０…検出部、５０…ライン演算部、６０、１０６０…配置演算部、７０、１０７０…再配置部、１０４０…標準書体フォントメモリ。

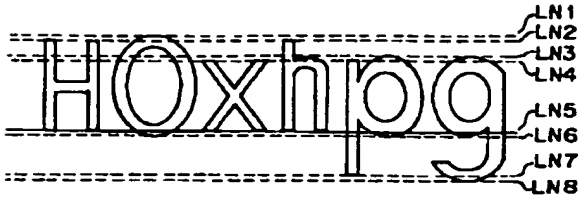
【図１】



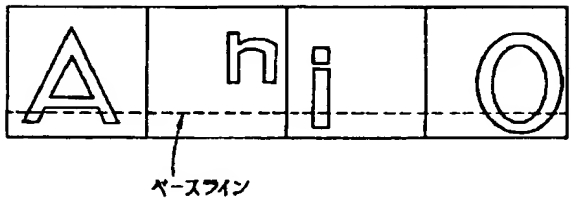
【図２】



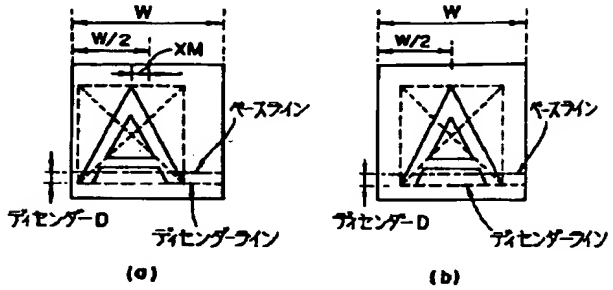
【図３】



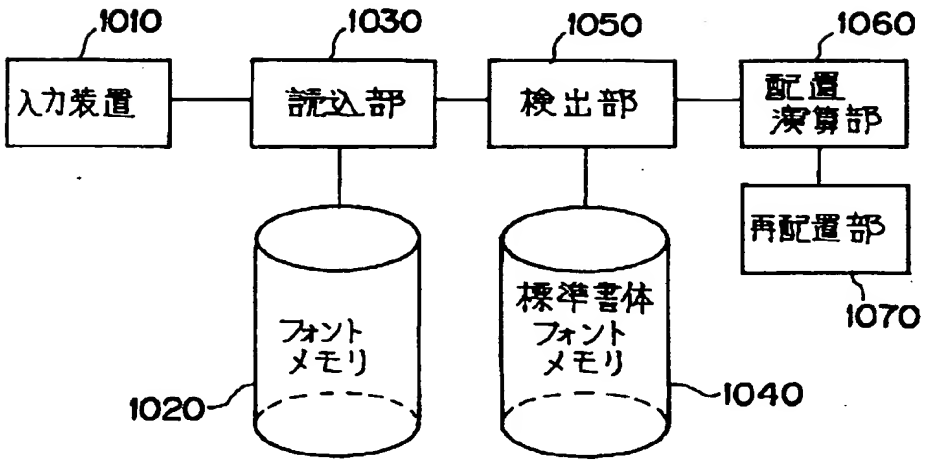
【図４】



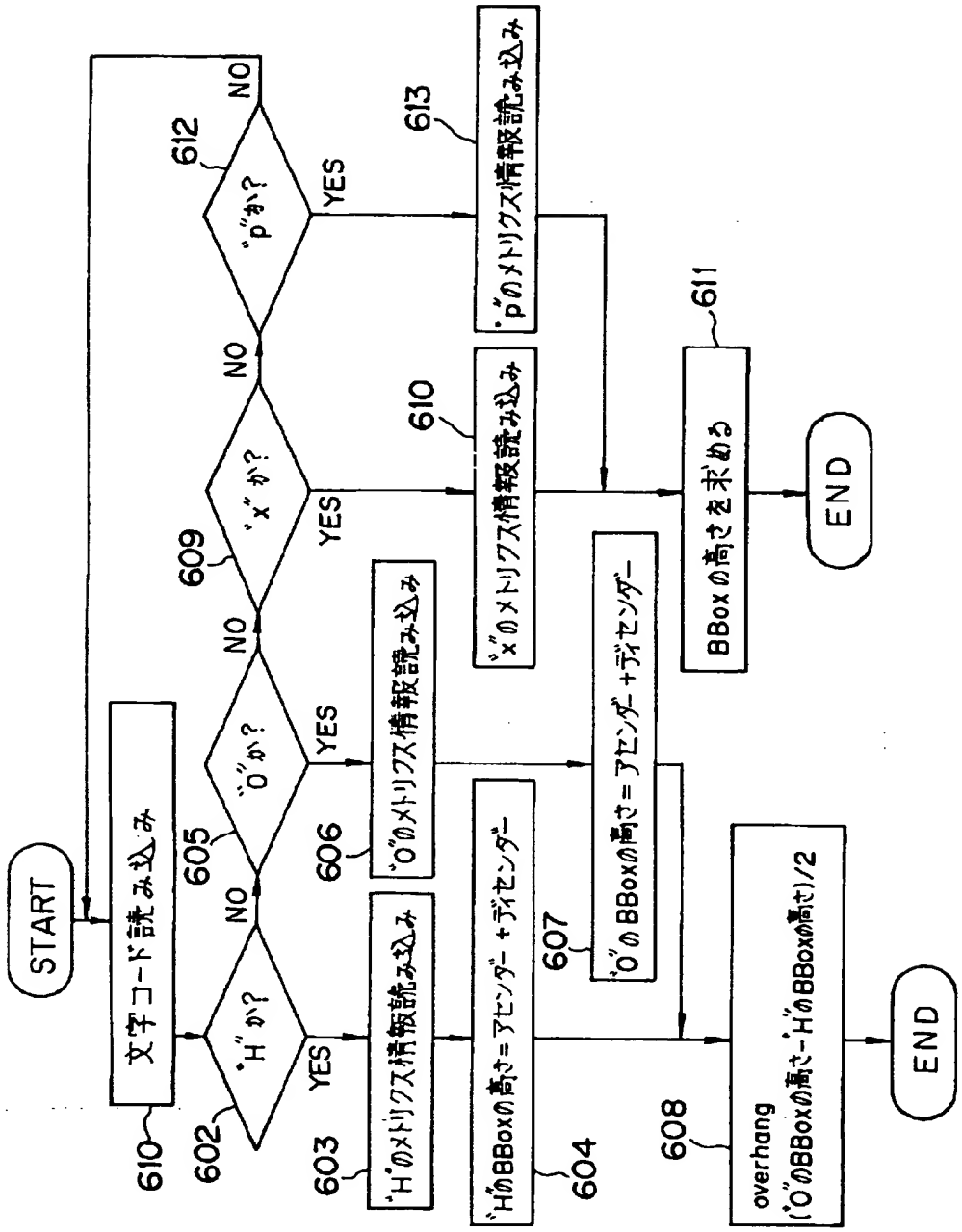
【図５】



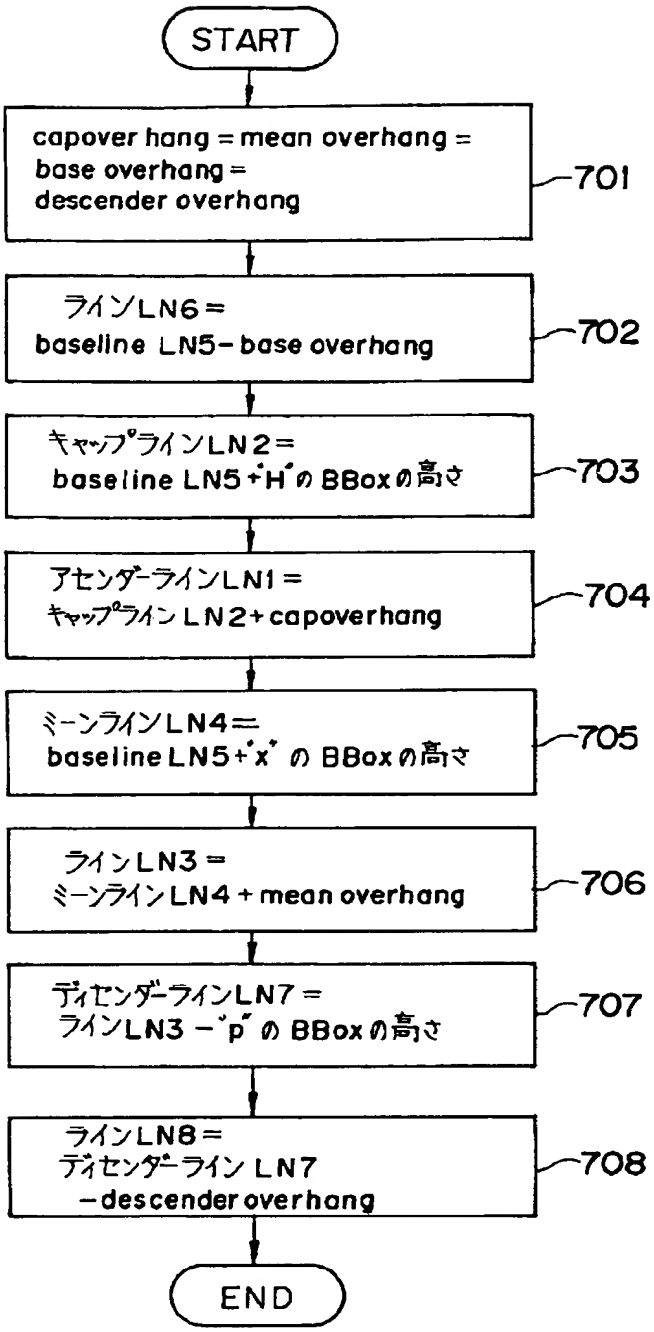
【図１０】



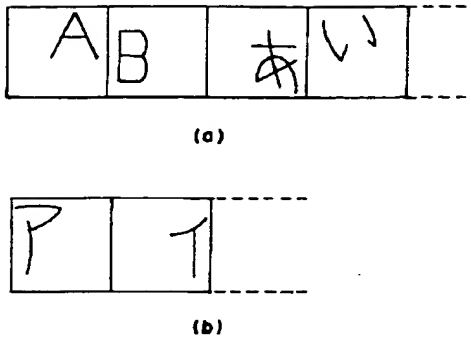
【図6】


START
文字コード読み込み
610
602
"H"か?
YES
603
"H"のメトリクス情報読み込み
604
"H"のBBoxの高さ=アセンダー+ディセンダー
NO
605
"O"か?
YES
606
"O"のメトリクス情報読み込み
607
"O"のBBoxの高さ=アセンダー+ディセンダー
NO
609
"x"か?
YES
610
"x"のメトリクス情報読み込み
NO
612
"p"か?
YES
613
"p"のメトリクス情報読み込み
608
overhang
("O"のBBoxの高さ-"H"のBBoxの高さ)/2
611
BBoxの高さを求める
END
END

【図７】

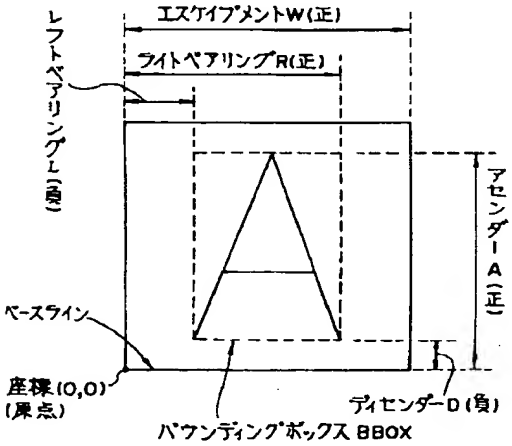


【図１１】

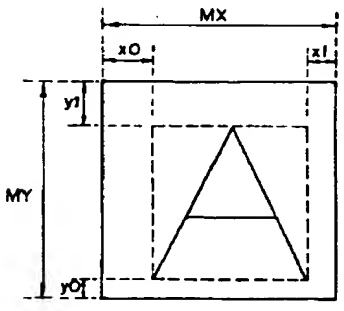


【図１２】

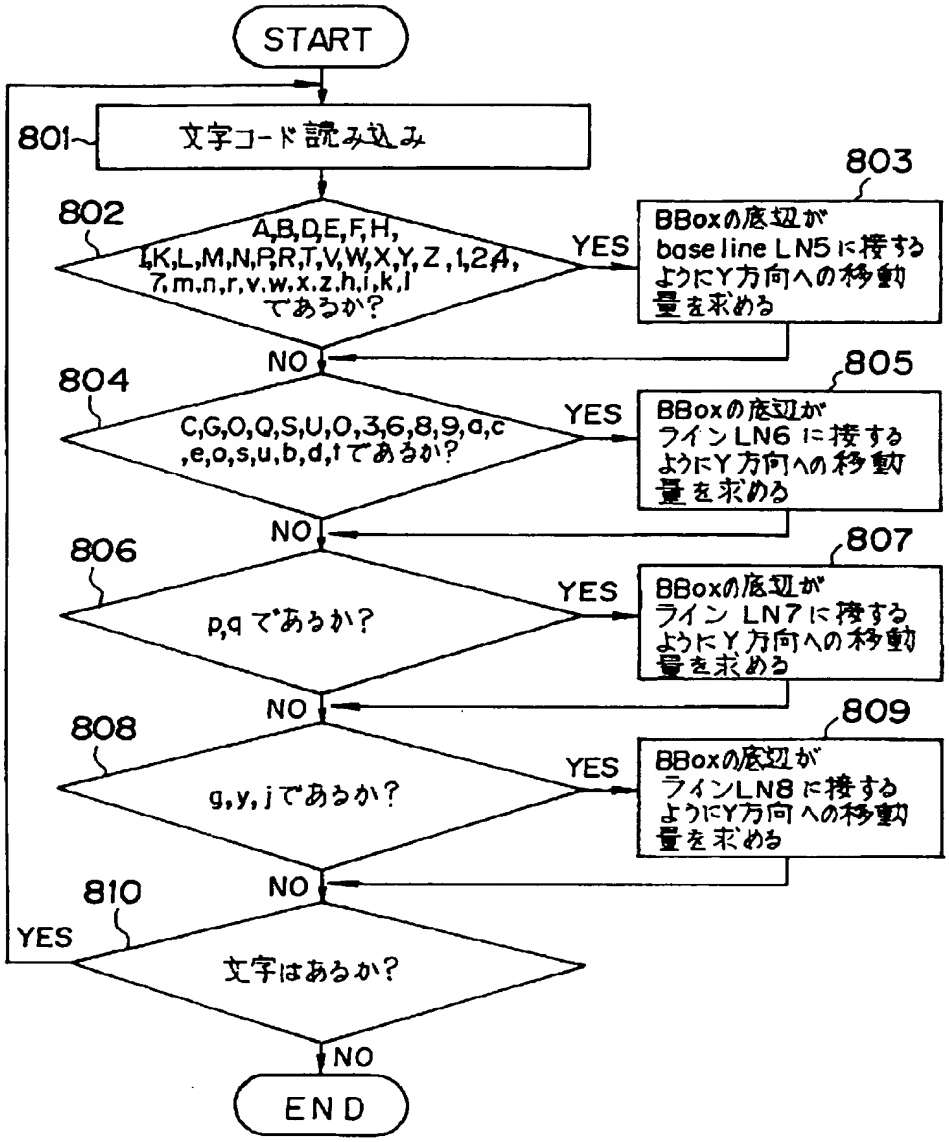
【図１３】



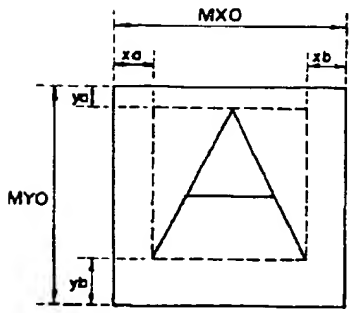
【図１４】



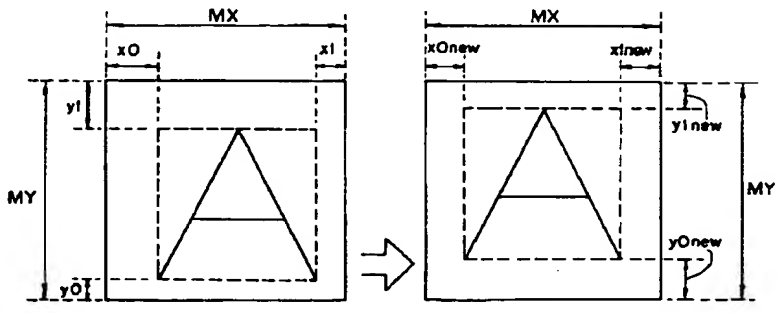
【図８】


START
801
文字コード読み込み
802
A,B,D,E,F,H,
I,K,L,M,N,P,R,T,V,W,X,Y,Z,1,2,4,
7,m,n,r,v,w,x,z,h,i,k,l
であるか?
YES
803
BBoxの底辺が
baseline LN5に接する
ようにY方向への移動
量を求める
NO
804
C,G,O,Q,S,U,0,3,6,8,9,a,c
,e,o,s,u,b,d,tであるか?
YES
805
BBoxの底辺が
ライン LN6 に接する
ようにY方向への移動
量を求める
NO
806
p,qであるか?
YES
807
BBoxの底辺が
ライン LN7に接する
ようにY方向への移動
量を求める
NO
808
g,y,jであるか?
YES
809
BBoxの底辺が
ライン LN8に接する
ようにY方向への移動
量を求める
NO
810
文字はあるか?
YES
NO
END

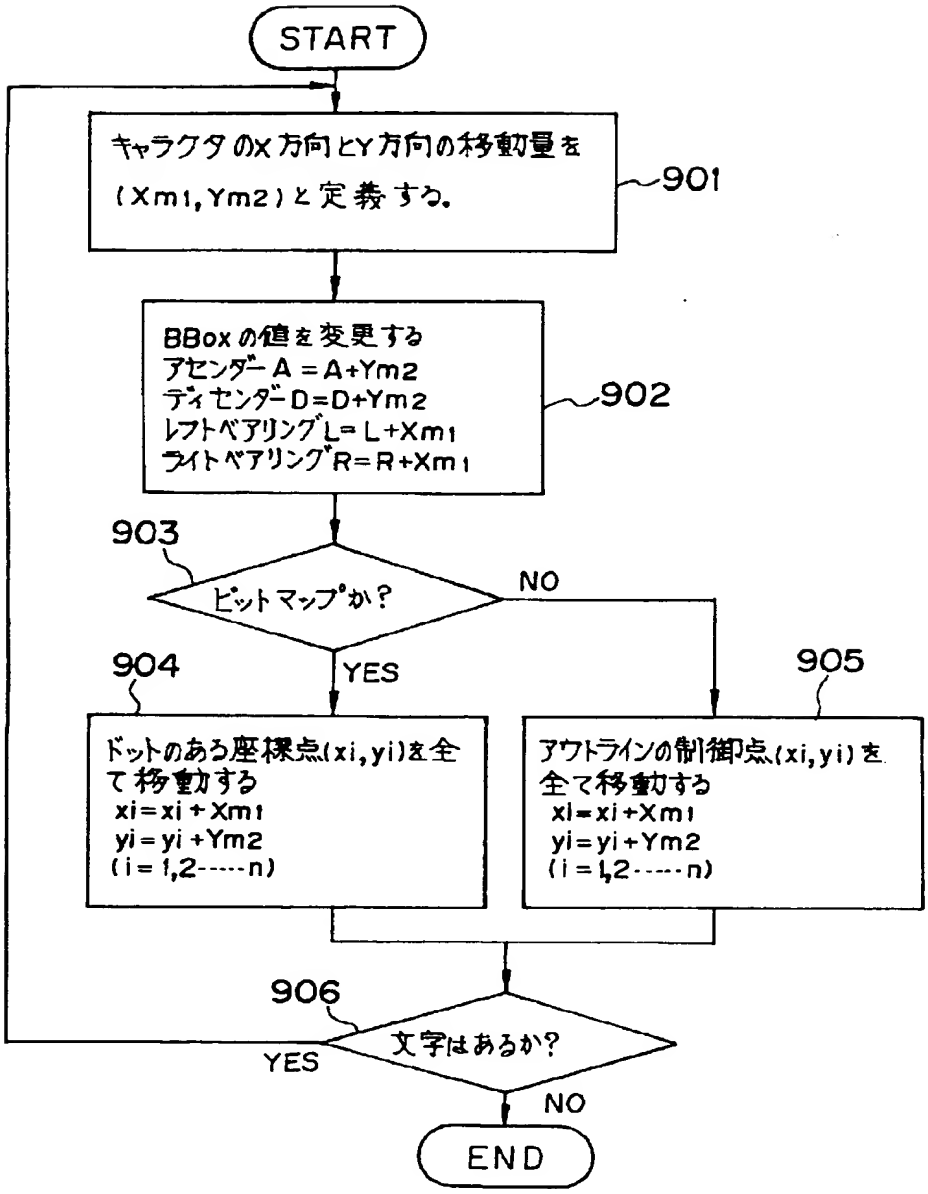
【図１５】


MXO
xa
xb
ya
MYO
yb

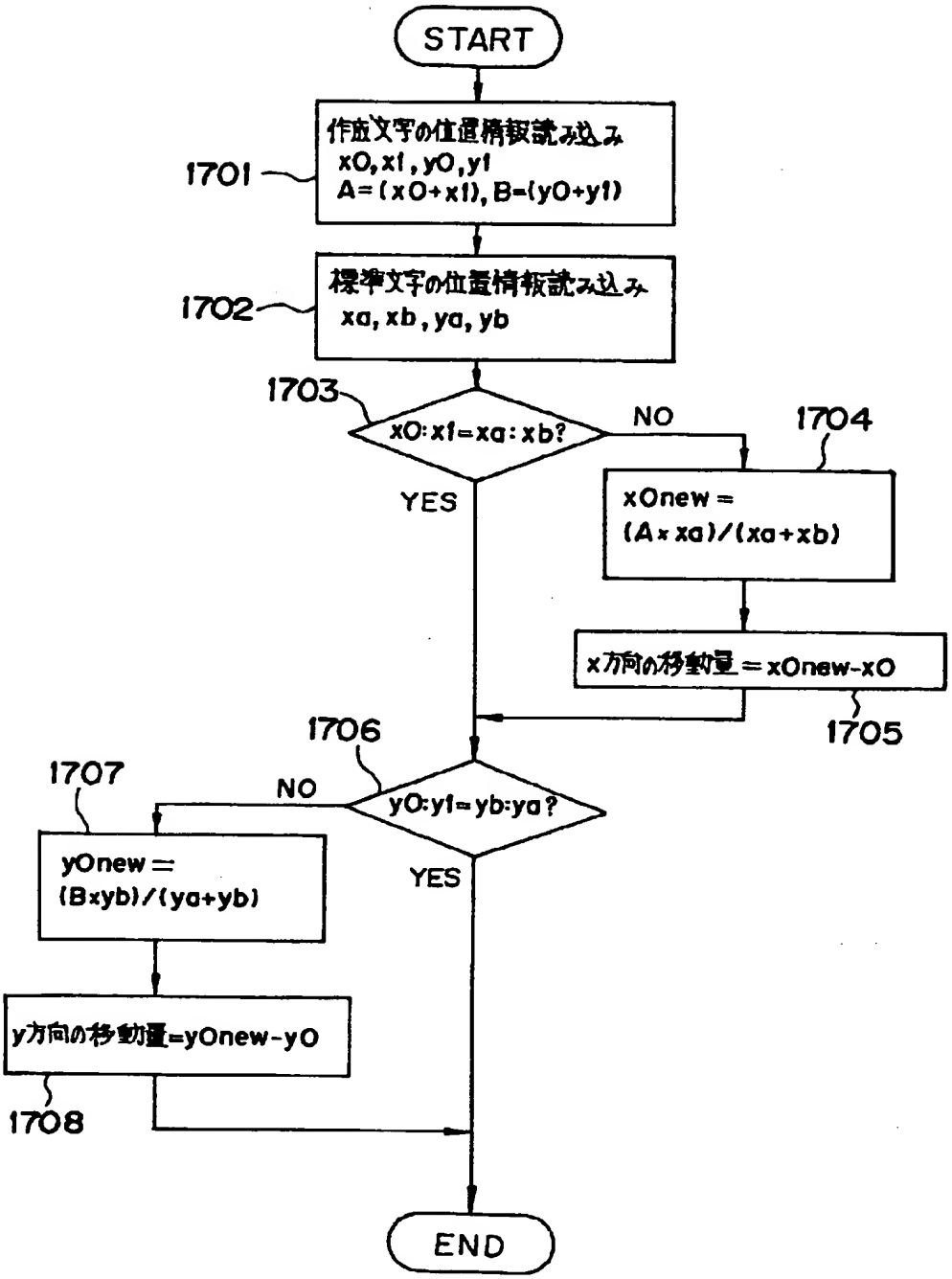
【図１６】


MX
xO
x1
y1
MY
yO
MX
xOnew
xlnew
ylnew
MY
yOnew

【図９】


START
キャラクタのX方向とY方向の移動量を
(Xm1,Ym2)と定義する。
901
BBoxの値を変更する
アセンダー A=A+Ym2
ディセンダー D=D+Ym2
レフトベアリング L=L+Xm1
ライトベアリング R=R+Xm1
902
903
ビットマップか?
NO
YES
904
905
ドットのある座標点(xi,yi)を全て移動する
xi=xi+Xm1
yi=yi+Ym2
(i=1,2-----n)
アウトラインの制御点(xi,yi)を全て移動する
xi=xi+Xm1
yi=yi+Ym2
(i=1,2-----n)
906
文字はあるか?
YES
NO
END

【図17】



フロントページの続き

(51)Int.Cl.[5]	識別記号	庁内整理番号	F I	技術表示箇所
B41J 2/485				
G06F 15/20	534	P 7343－5L		